Autonomous Smart Morphing Wing
Development, Realisation & Validation

Mkhoyan, T.

DOI
10.4233/uuid:b1c97841-fda7-420c-8feb-f1145faa531c

Publication date
2022

Document Version
Final published version

Citation (APA)
Mkhoyan, T. (2022). *Autonomous Smart Morphing Wing: Development, Realisation & Validation*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:b1c97841-fda7-420c-8feb-f1145faa531c

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Autonomous Smart Morphing Wing

Development, Realisation & Validation

Tigran Mkhoyan

Autonomous Smart Morphing Wing

Development, Realisation & Validation

Tigran Mkhoyan

Propositions

accompanying the dissertation

Autonomous Smart Morphing Wing

Development, Realisation & Validation

by

Tigran Mkhoyan

1. One should never attempt to solve a control hardware problem with software, particularly for a morphing wing.
2. Vision-based control approach provides high accuracy and high sampling capability required to cope with uncertainties of complex morphing mechanisms; therefore, it should be adopted as the preferred method for sensor-driven multi-objective control of actively morphing wings. *(this dissertation)*
3. The success of implementation for any sensor-driven advanced control objective for an autonomous morphing wing relies on reducing model dependency, increasing actuator bandwidth and facilitating hardware and software frameworks for data synchronisation. *(this dissertation)*
4. A distributed control surface system, which is ideally smooth and morphing, is a prerequisite for multi-objective control and aerodynamic performance optimisation. *(this dissertation)*
5. Conventional aircraft-design inevitably poses control problems. Control-design solves aircraft design and manufacturing uncertainties. The challenge for actively distributed morphing design is to solve all problems, *simultaneously. (this dissertation)*
6. The human species will truly advance when it has learned to fully embrace chaos rather than control it.
7. The trivial solution to restoring the equilibrium of the planets' ecosystem is removing the human from the equation. The challenge of the current century will be to arrive at a non-trivial solution, one that does not involve self-elimination.
8. The preposition "towards" is the scientific duct tape in academic writing.
9. Whatever can go wrong, *certainly* will go wrong during a wind tunnel experiment.
10. The overall rigorousness of implementation and analysis in complex software projects is measured by the wear of the ctrl button.

These propositions are regarded as opposable and defendable, and have been approved as such by the promotor dr. ir. R. De Breuker and promotor dr. ir. C.C. de Visser.

Stellingen

behorende bij het proefschrift

Autonomous Smart Morphing Wing

Development, Realisation & Validation

door

Tigran Mkhoyan

1. Er dient nooit te worden getracht een controle hardware probleem op te lossen met software, in het bijzonder niet bij een *morphing* vleugel.
2. Een op visuele controle benadering geeft een hoge mate van nauwkeurigheid en steekproefmogelijkheden welke weerstand kunnen bieden aan de onzekerheden van complexe *morphing* mechanismes; dit zou daarom moeten worden gekozen als de voorkeursmethode voor door sensor gedreven, multi-objective controle van actief *morphing* vleugels. *(dit proefschrift)*
3. Het succes van implementatie van elk door sensor-gedreven geavanceerde controledoel voor een autonoom *morphing* vleugel is afhankelijk van het verminderen van model-afhankelijkheid, toename van actuatorbandbreedte en het gebruik van onderliggende hardware- en softwaremethoden voor datasynchronisatie. *(dit proefschrift)*
4. Een verdeeld stuurvlaksysteem, welk idealiter glad en *morphing* is, is een voorwaarde voor multi-objective controle en aerodynamische prestatieoptimalisatie. *(dit proefschrift)*
5. Conventionele vliegtuigontwerpmethoden leiden onvermijdelijk tot controleproblemen. Controle-ontwerpmethoden bieden oplossingen voor onzekerheden in luchtvaartonwerpen en productie. Het is de uitdaging voor actieve, verdeelde, *morphing* ontwerpen om deze problemen simultaan op te lossen. *(dit proefschrift)*
6. De mensheid zal pas werkelijk vooruitgeraken wanneer het heeft geleerd de chaos volledig te omarmen in plaats van die te proberen te controleren.
7. De triviale oplossing om het evenwicht van ons ecosysteem te herstellen, is om de mens te verwijderen. De uitdaging van de huidige eeuw zal zijn om een non-triviale oplossing te vinden, welke niet leidt tot zelf-eliminatie.
8. Het voorzetsel "in de richting van" is het wetenschappelijke duct tape in academisch schrijven.
9. Alles wat fout kan gaan, zal zeker fout gaan tijdens een windtunnel experiment.
10. De algehele ontberingen tijdens de implementatie en analyse van complexe softwareprojecten kunnen worden gemeten aan de hand van de slijtage van de ctrl-toets.

Deze stellingen worden opponeerbaar en verdedigbaar geacht en zijn als zodanig goedgekeurd door de promotor dr. ir. R. De Breuker and promotor dr. ir. C.C. de Visser.

Autonomous Smart Morphing Wing

Development, Realisation & Validation

Autonomous Smart Morphing Wing

Development, Realisation & Validation

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 26 september 2022 om 12:30 uur

door

Tigran Mkhoyan

Ingenieur Luchtvaart en Ruimtevaart
Technische Universiteit Delft, Nederland,
geboren te Yerevan, Armenië.

Dit proefschrift is goedgekeurd door de:

promotor: dr. ir. R. De Breuker
promotor: dr. ir. C.C. de Visser

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
dr. ir. R. De Breuker	Technische Universiteit Delft, promotor
dr. ir. C.C. de Visser	Technische Universiteit Delft, promotor
Onafhankelijke leden:	
S. Blanc, M.S.c	Airbus
Prof. dr. C.E.S. Cesnik	University of Michigan
Prof. dr. F.J. Silvestre	Technische Universität Berlin
Prof. dr. ir. J.W. van Wingerden	Technische Universiteit Delft
Prof. dr. F. Scarano	Technische Universiteit Delft

Keywords: morphing design, control, sensing, over-actuated system

Printed by: Ipskamp Printing

Front & Back: Tigran Mkhoyan

ISBN 978-94-6421-868-8

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Contents

Preface

Ex nihilo nihil fit. "Nothing comes out of nothing" or "out of nothing, nothing is produced", are the words that precede the body of this writing, as the reader will soon realise. This phrase, believed to have its origin in the writings of Parmenides and later adopted by Lucretius, leaves room for various interpretations. For one, it can be interpreted as the effort we must exert through action for a desirable outcome to materialise. On a more philosophical note, forgoing the weighty theoretical discussion, this phrase may invite a debate on the universe's existence, the creation of matter and the subtleties of the quantum vacuum.

Returning to the subject of problem-solving, should we focus our efforts on the intuitive solution rather than exploring a myriad of parallel directions to obtain a better understanding of the problem and its implications? Perhaps it is both. First, we must bend our thoughts and imaginations with great flexibility, then reflect and create.

So, shall we let our imagination[1] run wild for a moment and see if a new world of possibilities might emerge?

With this in mind, let us embark on the subject of the dissertation, which the author hopes the reader shall enjoy.

Tigran Mkhoyan
Lucerne, September 2022

Let us imagine *"An aircraft from the future with a bio-inspired wing, in hyperrealistic detail".*

[1] The following pictures were created by translating the quoted text into an AI-generated picture.

Summary

With the increasing desire of the aerospace industry to reduce emissions and fuel consumption, morphing wings have gained much interest due to the ability to adapt the wing shape in-flight for improved energy efficiency and aerodynamic performance. *Active wing morphing* is a technology that can improve the aerodynamic performance continuously through different flight phases. However, a multidisciplinary approach is needed considering the design, modelling, control and integration of multi-objective control and sensing framework to develop a smart autonomous morphing wing system capable of adapting its shape autonomously. The SmartX project was initiated for this purpose at the Delft University of Technology, Faculty of Aerospace Engineering, Department of Aerospace Structures and Materials, aiming to investigate the energy-efficient wing concepts through smart wings.

This dissertation presents *the Development, Realisation & Validation* of a smart morphing wing, the SmartX-Alpha, capable of meeting various real-time objectives with distributed seamless morphing modules. This is done through a holistic approach considering all building blocks of a morphing system presented in four Parts of the dissertation covering sensing, modelling, control, design and manufacturing aspects.

Part I, tackles the sensing approach required to reconstruct the shape of the wing in real-time. For this purpose, a vision-based sensing approach for morphing and very flexible wings is proposed, suitable for real-time implementation in a multi-objective control scheme in a non-invasive manner. A wind tunnel experiment was conducted with a very flexible wing to validate this approach in three individual studies investigated in the Open Jet Facility (OJF).

First, a non-invasive vision-based image tracking pipeline was developed using a robust machine learning approach to automatically detect and label visual markers from an image stream and integrate state estimation routing in the control feedback loop. An emphasis was put on robustness and the ability of the algorithm to deal with image noise. A novel approach was proposed using an inverse formulation of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN), the inverse DBSCAN^{-1}, where the clustering problem is reformulated into a noise filtering problem. Instead of *rejecting* the noise, this approach *explicitly detects* the noise, making the clustering an *implicit* task. The experimental dataset was processed using the DBSCAN^{-1} pipeline, and it was shown that the actual clusters were successfully identified and isolated from the noise in the image.

Thereafter, full shape reconstruction, utilising the earlier sorting and detection methodology, was studied. The primary purpose of the shape reconstruction method was to develop a robust state reconstruction approach, which would take as input a point correspondence pair detected from two image frames and reconstruct the tip deflection response in 3D induced by gust input on the wing. The proposed method consisted of a tracker pair composed of a purely visual filter, a high-speed Kernelised Correlation Filter (KCF), paired with an Extended Kalman Filter (EKF), allowing robust estimation of a system exhibiting

oscillatory motion under the presence of marker failure and occlusions. The KCF-EKF method was validated experimentally on a real-time image stream of a very flexible wing subjected to gust excitation in the Open Jet Facility (OJF) wind tunnel. The method is further improved with the expansion of the EKF to the Augmented Extended Kalman Filter (AEKF) form, where the uncertain system parameters are included in the Kalman Filter model, thereby estimating both the states and parameters online adaptively. The last stage of the reconstruction step is the triangulation with the Direct Linear Transformation (DLT) method and coordinate transformation needed to obtain the relative displacement from the static displacement under a given free stream velocity.

The last study of this Part investigated how the introduction of artificial intelligence, in particular deep learning from raw image data, could help simplify the processes involved in the proposed smart visual sensing approach or even potentially replace it entirely. To explore the deep learning-based concept of *Smarter Visual Sensing*, a Deep Convolutional Neural Network (DCNN) is trained to perform a part of the sorting operation and detect the orientation angle of extracted contour shape of the wing. Simultaneously, the Geometrical Reflectional Symmetry Detector (GeConv) algorithm proposes an efficient algorithm-based solution to the problem to compare the two methods. Both approaches show excellent results in detecting symmetry and orientation. The Neural Network can achieve an angle error of up to 0.05 degrees without needing a contour shape as an input.

Part II presents the design, development, realisation and experimental testing of a novel distributed modular morphing concept, SmartX-Alpha. Furthermore, a learning-based shape optimisation approach is investigated, aiming to investigate the benefits of distributed morphing design combined with online shape optimisation.

The distributed morphing concept is integrated into a seamless, actively morphing wing demonstrator called SmartX-Alpha. The proposed concept allowed variation of lift distribution locally along the span and addressed the drawbacks of the initial Translation Induced Camber (TRIC) concept. An Fluid-Structure Interaction (FSI) optimisation framework was developed to produce the optimised laminated design of the morphing skin, taking into account the ply orientation, laminate thickness, laminate properties and actuation loads. Fixed ± 45 degree ply orientation and gradual ply thickness dropping achieved the desired morphing flexibility and manufacturability. A gap fully covered by silicone skin yielded the best design configuration for the inter-modular skin, allowing sufficient flexibility and having the most negligible impact on the actuator load. The morphing shapes are assessed, and mechanical imperfections, contributed mostly by backlash hysteresis, are characterised in experimental validation with Digital Image Correlation (DIC). This effect is later addressed by a compensating vision-based control approach in the subsequent Part.

Next, a numerical study is performed to assess the aerodynamic performance gain that could be achieved with the proposed seamless, distributed morphing design, utilising artificial intelligence as a black-box controller. For this purpose, a novel learning-based shape optimisation method was proposed to optimise the shape online such that the steady-state lift-to-drag ratio could be optimised for a given target lift coefficient using the lift and drag measurement updates. An online-trained Artificial Neural Network (ANN) based on Radial Basis Function Neural Network (RBF-NN) acted as an onboard lift and drag coefficient value model for given actuator input angles, which could be adapted online. An

evolutionary optimisation algorithm was responsible for finding the most optimal shape and the angle of attack for the given lift coefficient targets. The simulation experiment demonstrated that the surrogate model could estimate sufficiently across a large spectrum of inputs and outputs, facilitating the optimiser to produce optimal wing shapes outperforming the jig shape in terms of aerodynamic efficiency. With the global nature of the optimisation approach, the optimisation strategy allowed to find wing shape and angle-of-attack combinations with lift-to-drag ratio increases of up to 14.6 % for a wide range of target lift coefficients without requiring further exploration. The ability to update the onboard Radial Basis Function Neural Network (RBF-NN) surrogate model using new measurements demonstrated a relevant and beneficial aspect of the proposed approach to actively morphing systems, namely, *adaptability*.

Part III presents the multi-objective control laws developed to simultaneously meet the gust and manoeuvre load alleviation objective and the real-time shape optimisation strategy to improve online aerodynamic performance. Furthermore, a vision-based control strategy is proposed to mitigate nonlinearities in the actuation system arising from mechanical imperfections. A series of wind tunnel experiments are conducted in the OJF to validate the methodologies on the SmartX-Alpha, ensuring the objectives are satisfied autonomously, in-real time.

First, a sensor-driven, incremental control method is proposed for simultaneous gust and manoeuvre load alleviation combining Incremental Non-linear Dynamic Inversion (INDI) and Quadratic Programming (QP). Since the morphing system has a distributed array of actuators, the INDI is combined with quadratic programming (INDI-QP) to achieve a control allocation strategy, which considers the actuator position, rate and relative position constraints. To increase the smoothness of the control allocation, the INDI-QP is augmented with the virtual shape functions (denoted as INDI-QP-V). The control approach is paired with an integrated distributed control system architecture allowing efficient and real-time processing of the distributed morphing wing system. The effectiveness of the proposed INDI-QP-V is validated by wind-tunnel experiments in a complex experimental setup running in real-time. In manoeuvre load alleviation tasks, INDI-QP-V increased the total lift for pulling-up manoeuvres without amplifying the wing root bending moment. In the presence of successive "1-cos" gusts, INDI-QP-V mitigated the loads without requiring any gust information. A key revelation of the proposed method was the robustness against aerodynamic uncertainties, gusts, actuator faults, and nonlinear backlash. The INDI-QP-V could mitigate the effect of gust, while the 9^{th} actuator had become inoperable due to mechanical failure.

Following this, a complex experimental system is designed where all previous elements of an active morphing wing are combined to investigate online performance optimisation. Before forwarding this aim, the backlash hysteresis in the system is addressed using a vision-based data-driven control approach. A vision-based control system was developed to provide accurate knowledge of the morphing wings' shape to the controller without invasive changes to the morphing system. The controller utilises an Incremental Model-based Nonlinear Dynamic Inversion (IM-NDI) control approach, allowing the system dynamics to be identified online using the stored input/output data. The experimental results demonstrate that the morphing wing can track reference signals with different frequencies by applying the IM-NDI despite external disturbances. Under Feed-Forward (FF) control,

the morphing wing suffers from mechanical imperfections, reflected by the tracking responses' lagging and magnitude shrinking phenomena. The performance of feed-forward control also degrades in the presence of external disturbances. By contrast, experimental results show that IM-NDI can effectively decrease the tracking errors by more than 62 % despite disturbances.

Finally, the shape optimisation objective is addressed, demonstrating a novel learning-based aerodynamic performance optimisation approach integrated into a complex real-time experimental system. Various key components of the morphing wing concept are validated in this final wind tunnel experiment in OJF. The proposed approach consists of an evolutionary optimisation strategy, Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) coupled with a low cost, RBF-NN onboard surrogate model for online optimisation of the steady-state lift-to-drag ratio of the morphing wing for a given target lift coefficient. The onboard RBF-NN model is trained with experimental data collected during wandering phases to capture the behaviour of the wing model and predict the steady-state lift and drag coefficients. Compared to the unmorphed wing base shape, a drag reduction of 7.8 % was achieved on the SmartX-Alpha demonstrator for a target lift coefficient of 0.65. For a wide range of target lift coefficients, the predicted drag reductions vary between 6.5 % and 19.8 %, with higher drag reductions associated with lower lift coefficients. The method demonstrated that online real-time performance optimisation could be achieved globally with an adaptive onboard model. High-accuracy and fast sampling sensors were critical for backlash compensation and shape optimisation, provided by vision-based sensing.

The final Part, Part IV, is focused on the prospective outlook of over-actuated distributed smart morphing wing concepts, their maturing and adoption in new aircraft designs.

Throughout the previous Parts, the active morphing wing demonstrator concept was proposed in the previous sections, and its feasibility was investigated through incremental improvements and successive wind tunnel tests. A key finding was that actuator bandwidth, and the quality of the sensor noise attenuation were critical factors required to push the envelope for multi-objective optimisation and mitigation of aeroelastic loads. The last Part investigates the impact of faster actuation on load alleviation performance and the ability to achieve faster aeroelastic objectives. For this purpose, a distributed over-actuated aeroelastic wing demonstrator was developed, the SmartX-Neo, allowing faster actuation with a much simpler flap mechanism and ensuring the same integrated sensing capability as the SmartX-Alpha.

The effect of actuator design was studied through a series of gust simulations of closed-loop control of a parametric aeroservoelastic model of the SmartX-Neo, for gusts with various frequencies. Actuator dynamics were implemented in the system by scaling the stiffness and damping of a second-order system. With the nominal Linear Quadratic Regulator (LQR) controller, reductions of peak gust load up to 78 % were achieved compared to the open-loop case. It was observed that lower scaling corresponding to faster actuation provides significant improvements of up to 11 % over the nominal actuator configuration, yielding reductions of gust loads up to 84 %. Furthermore, it was observed that the effectiveness of faster actuators improves for higher frequency gusts, confirming the potential of SmartX-Neo to deal with faster control objectives more effectively.

Samenvatting

Met de toenemende wens van de lucht- en ruimtevaartindustrie om emissies en brandstofverbruik te verminderen, hebben *morphing* vleugels veel belangstelling gekregen vanwege het vermogen om de vleugelvorm tijdens de vlucht aan te passen voor betere energie-efficiëntie en aerodynamische prestaties. *Active wing morphing* is een technologie die de aerodynamische prestaties tijdens verschillende vluchtfasen continu kan verbeteren. Er is echter een multidisciplinaire aanpak nodig, rekening houdend met het ontwerp, de modellering, de controle en de integratie van een multi-objectief controle- en detectiekader om een slim autonoom *morphing* vleugelsysteem te ontwikkelen dat in staat is zijn vorm autonoom aan te passen. Hiervoor is het SmartX-project geïnitieerd aan de TU Delft, faculteit Luchtvaart- en Ruimtevaarttechniek, afdeling Aerospace Structures and Materials, met als doel de energiezuinige vleugelconcepten door middel van slimme vleugels te onderzoeken.

Dit proefschrift presenteert *de ontwikkeling, realisatie & validatie* van een slimme *morphing* vleugel, de *SmartX-Alpha*, die in staat is om verschillende real-time doelstellingen te bereiken met gedistribueerde naadloze *morphing* modules. Dit wordt gedaan door middel van een holistische benadering waarbij alle bouwstenen van een *morphing* systeem worden onderzocht, gepresenteerd in vier delen van het proefschrift, die detectie, modellering, controle, ontwerp en fabricageaspecten behandelen.

Deel I behandelt de detectiebenadering die nodig is om de vorm van de vleugel in *real-time* te reconstrueren. Voor dit doel wordt een op visie gebaseerde detectiebenadering voor morphing en zeer flexibele vleugels voorgesteld, geschikt voor *real-time* implementatie in een multi-objectief controleschema op een niet-invasieve manier. Een windtunnelexperiment werd uitgevoerd met een zeer flexibele vleugel om deze benadering te valideren in drie afzonderlijke onderzoeken die in de *Open Jet Facility (OJF)* werden onderzocht.

Ten eerste werd een niet-invasieve, op visie gebaseerde pijplijn voor het volgen van afbeeldingen ontwikkeld met behulp van een robuuste *machine learning* benadering om automatisch visuele markeringen uit een beeldstroom te detecteren en te labelen en om statusschattingsroutering te integreren in de controle-feedbacklus. De nadruk werd gelegd op robuustheid en het vermogen van het algoritme om met beeldruis om te gaan. Een nieuwe benadering met behulp van een inverse formulering van de *Density-Based Spatial Clustering of Applications with Noise (DBSCAN)*, de inverse DBSCAN^{-1}, werd voorgesteld waarbij het clusteringsprobleem wordt geherformuleerd in een ruisfilterprobleem. In plaats van de ruis *verwerpen*, detecteert deze benadering *expliciet* de ruis, waardoor het clusteren een *impliciete* taak wordt. De experimentele dataset werd verwerkt met behulp van de DBSCAN^{-1}-pijplijn, en er werd aangetoond dat de daadwerkelijke clusters met succes werden geïdentificeerd en geïsoleerd van de ruis in de afbeelding.

Dit wordt gevolgd door een volledige vormreconstructiemethode, waarbij gebruik wordt gemaakt van de eerdere sorteer- en detectiemethode. Het primaire doel van de vormreconstructiemethode was om een robuuste toestandsreconstructiebenadering te ontwikkelen,

die als invoer een puntcorrespondentiepaar zou nemen dat werd gedetecteerd uit twee beeldframes en de tipafbuigingsrespons in 3D zou reconstrueren, veroorzaakt door windvlaaginvoer op de vleugel.

De voorgestelde methode bestond uit een trackerpaar bestaande uit een puur visueel filter, een hogesnelheids *Kernelised Correlation Filter (KCF)*, gecombineerd met een *Extended Kalman Filter (EKF)*, wat een robuuste schatting mogelijk maakt van een systeem dat oscillerende beweging vertoont onder de aanwezigheid van markerfalen en occlusies. De KCF-EKF-methode werd experimenteel gevalideerd op een *real-time* beeldstroom van een zeer flexibele vleugel onderworpen aan windvlaagexcitatie in de OJF windtunnel. De methode is verder verbeterd met de uitbreiding van de EKF naar de *Augmented Extended Kalman Filter (AEKF)* vorm, waarbij de onzekere systeemparameters zijn opgenomen in het Kalman Filter-model, waardoor zowel de toestanden als parameters adaptief online worden geschat. De laatste fase van de reconstructiestap is de triangulatie met de *Direct Linear Transformation (DLT)* methode en coördinaattransformatie die nodig is om de relatieve verplaatsing van de statische verplaatsing onder een gegeven vrije stroomsnelheid te verkrijgen.

De laatste studie van dit deel onderzocht hoe de introductie van kunstmatige intelligentie, met name *deep learning* uit onbewerkte beeldgegevens, kan helpen de processen die betrokken zijn bij de voorgestelde slimme visuele waarnemingsaanpak te vereenvoudigen of deze mogelijk zelfs volledig te vervangen. Om het op *deep learning* gebaseerde concept van *Smarter Visual Sensing* te verkennen, wordt een *Deep Convolutional Neural Network (DCNN)* getraind om een deel van de sorteerbewerking uit te voeren en de oriëntatiehoek van de geëxtraheerde contourvorm van de vleugel te detecteren. Tegelijkertijd stelt het GeConv-algoritme (GeConv) een efficiënte, op algoritmen gebaseerde oplossing voor het probleem voor om de twee methoden te vergelijken. Beide benaderingen laten uitstekende resultaten zien bij het detecteren van symmetrie en oriëntatie. Het neurale netwerk kan een hoekfout van maximaal 0.05 graden bereiken zonder dat een contourvorm als invoer nodig is.

Deel II presenteert het ontwerp, de ontwikkeling, de realisatie en het experimenteel testen van een nieuw gedistribueerd modulair *morphing* concept, SmartX-Alpha. Verder wordt een op leren gebaseerde vormoptimalisatiebenadering onderzocht, met als doel de voordelen van gedistribueerd *morphing* ontwerp in combinatie met online vormoptimalisatie te onderzoeken.

Het gedistribueerde *morphing* concept is geïntegreerd in een naadloze, actief morphing wing-demonstrator genaamd SmartX-Alpha. Het voorgestelde concept maakte variatie van de liftdistributie lokaal langs de overspanning mogelijk en loste de nadelen van het oorspronkelijke *Translation Induced Camber (TRIC)* concept op. Een *Fluid-Structure Interaction (FSI)* optimalisatieraamwerk werd ontwikkeld om het geoptimaliseerde gelamineerde ontwerp van de morphing skin te produceren, rekening houdend met de laagoriëntatie, laminaatdikte, laminaateigenschappen en activeringsbelastingen. Een vaste laagoriëntatie van ± 45 graden en een geleidelijke daling van de laagdikte zorgden voor de gewenste *morphing* flexibiliteit en maakbaarheid. Een opening volledig bedekt door siliconenhuid leverde de beste ontwerpconfiguratie voor de intermodulaire huid op, waardoor voldoende flexibiliteit mogelijk was en de meest verwaarloosbare impact op de actuatorbelasting. De *morphing* vormen worden beoordeeld en mechanische onvolkomenheden,

voornamelijk veroorzaakt door spelinghysterese, worden gekarakteriseerd in experimentele validatie met DIC. Dit effect wordt later aangepakt door een compenserende, op visie gebaseerde regelaanpak in het volgende deel.

Vervolgens wordt een numerieke studie uitgevoerd om de aerodynamische prestatiewinst te beoordelen die kan worden bereikt met het voorgestelde naadloze, gedistribueerde *morphing* ontwerp, waarbij kunstmatige intelligentie wordt gebruikt als een *black-box controller*. Voor dit doel werd een nieuwe, op leren gebaseerde vormoptimalisatiemethode voorgesteld om de vorm online te optimaliseren, zodat de steady-state lift-naar-weerstandsverhouding kan worden geoptimaliseerd voor een bepaalde doelliftcoëfficiënt met behulp van de updates voor lift- en weerstandsmetingen. Een online getraind kunstmatig neuraal netwerk (ANN) op basis van radiale basisfuncties neurale netwerkstructuur (RBF-NN) fungeerde als een *onboard* lift- en weerstandscoëfficiëntwaardemodel voor gegeven actuatorinvoerhoeken, die online konden worden aangepast. Een evolutionair optimalisatie-algoritme was verantwoordelijk voor het vinden van de meest optimale vorm en de aanvalshoek voor de gegeven liftcoëfficiëntdoelen. Het simulatie-experiment toonde aan dat het surrogaatmodel voldoende kon schatten over een groot spectrum van inputs en outputs, waardoor de optimizer optimale vleugelvormen kon produceren die beter presteerden dan de malvorm in termen van aerodynamische efficiëntie. Met het globale karakter van de optimalisatiebenadering, maakte de optimalisatiestrategie het mogelijk om combinaties van vleugelvorm en aanvalshoek te vinden met verhogingen van de lift-naar-sleepverhouding tot 14.6 % voor een breed scala aan doelliftcoëfficiënten zonder verdere noodzaak verkenning. De mogelijkheid om het ingebouwde RBF-NN-surrogaatmodel bij te werken met behulp van nieuwe metingen, toonde een relevant en gunstig aspect aan van de voorgestelde benadering van actieve *morphing* systemen, namelijk *adaptability*.

Deel III presenteert de multi-objectieve controlewetten die zijn ontwikkeld om tegelijkertijd te voldoen aan de doelstelling voor het verlichten van windstoten en manoeuvres en de real-time vormoptimalisatiestrategie om de online aerodynamische prestaties te verbeteren. Verder wordt een op visie gebaseerde regelstrategie voorgesteld om niet-lineariteiten in het aandrijfsysteem als gevolg van mechanische onvolkomenheden te verminderen. In het OJF wordt een reeks windtunnelexperimenten uitgevoerd om de methodologieën op de SmartX-Alpha te valideren, zodat de doelstellingen autonoom en in *real-time* worden bereikt.

Ten eerste wordt een sensorgestuurde, incrementele regelmethode voorgesteld voor gelijktijdige verlichting van windstoten en manoeuvres door INDI en QP te combineren. Aangezien het *morphing* systeem een gedistribueerde reeks actuatoren heeft, is de INDI gecombineerd met kwadratische programmering (INDI-QP) om een besturingstoewijzingsstrategie te bereiken, die rekening houdt met de actuatorpositie, snelheid en relatieve positiebeperkingen. Om de soepelheid van de besturingstoewijzing te vergroten, is de INDI-QP uitgebreid met de virtuele vormfuncties (aangeduid als INDI-QP-V). De besturingsbenadering is gekoppeld aan een geïntegreerde architectuur van het gedistribueerde besturingssysteem, waardoor een efficiënte en *real-time* verwerking van het gedistribueerde *morphing* vleugelsysteem mogelijk is. De effectiviteit van de voorgestelde INDI-QP-V wordt gevalideerd door windtunnelexperimenten in een complexe experimentele opstelling die in *real-time* draait. Bij het verlichten van manoeuvrebelastingen verhoogde INDI-QP-V de totale lift voor optrekmanoeuvres zonder het buigmoment van de

vleugelwortel te vergroten. In aanwezigheid van opeenvolgende "1-cos" windstoten, heeft INDI-QP-V de belastingen verminderd zonder dat er windvlaaginformatie nodig was. Een belangrijke onthulling van de voorgestelde methode was de robuustheid tegen aerodynamische onzekerheden, windstoten, actuatorfouten en niet-lineaire speling. De INDI-QP-V kon het effect van windvlagen verminderen, terwijl de negende actuator onbruikbaar was geworden door een mechanisch defect.

Hierna wordt een complex experimenteel systeem ontworpen waarin alle eerdere elementen van een actieve *morphing* vleugel worden gecombineerd om online prestatie-optimalisatie te onderzoeken. Alvorens dit doel door te realiseren, wordt de spelinghysterese in het systeem aangepakt met behulp van een op visie gebaseerde, gegevensgestuurde besturingsbenadering. Er is een op zicht gebaseerd besturingssysteem ontwikkeld om nauwkeurige kennis van de vorm van de *morphing* vleugels aan de regelaar te verstrekken zonder invasieve veranderingen aan het morphing-systeem. De regelaar maakt gebruik van een incrementele, op modellen gebaseerde, niet-lineaire dynamische inversie (IM-NDI) regelbenadering, waardoor de systeemdynamiek online kan worden geïdentificeerd met behulp van de opgeslagen invoer-/uitvoergegevens. De experimentele resultaten tonen aan dat de *morphing* vleugel referentiesignalen met verschillende frequenties kan volgen door de IM-NDI toe te passen ondanks externe storingen. Onder controle van *Feed-Forward (FF)* heeft de *morphing* vleugel te lijden van mechanische onvolkomenheden, weerspiegeld door de achterblijvende en magnitudekrimpende verschijnselen van de trackingresponsen. De prestaties van de *feed-forward* regelaar nemen ook af in aanwezigheid van externe verstoringen. Daarentegen laten experimentele resultaten zien dat IM-NDI ondanks storingen de tracking errors met meer dan $62\ \%$ kan verminderen.

Ten slotte wordt het doel van vormoptimalisatie aangepakt, waarbij een nieuwe, op leren gebaseerde, aerodynamische prestatie-optimalisatiebenadering wordt gedemonstreerd, geïntegreerd in een complex *real-time* experimenteel systeem. Verschillende belangrijke componenten van het *morphing* vleugelconcept worden gevalideerd in dit laatste windtunnelexperiment in OJF.

De voorgestelde aanpak bestaat uit een evolutionaire optimalisatiestrategie, *Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES)* gekoppeld aan een goedkoop RBF-NN *onboard* surrogaatmodel voor online optimalisatie van de stationaire lift naar sleepverhouding van de *morphing* vleugel voor een gegeven doelliftcoëfficiënt. Het ingebouwde RBF-NN-model is getraind met experimentele gegevens die zijn verzameld tijdens zwervende fasen om het gedrag van het vleugelmodel vast te leggen en de stationaire lift- en luchtweerstandscoëfficiënten te voorspellen. Vergeleken met de niet-vervormde vorm van de vleugelbasis werd een weerstandsvermindering van $7.8\ \%$ bereikt op de SmartX-Alpha demonstrateur voor een beoogde liftcoëfficiënt van 0.65. Voor een groot aantal beoogde liftcoëfficiënten variëren de voorspelde weerstandsverminderingen tussen $6.5\ \%$ en $19.8\ \%$, met hogere weerstandsverminderingen geassocieerd met lagere liftcoëfficiënten. De methode toonde aan dat online *real-time* prestatie-optimalisatie wereldwijd kan worden bereikt met een adaptief *onboard* model. Zeer nauwkeurige en snelle bemonsteringssensoren waren van cruciaal belang voor spelingcompensatie en vormoptimalisatie, geleverd door Visie-gebaseerde detectie.

Het laatste deel, Deel IV, is gericht op de toekomstige vooruitzichten van overmatig geactiveerde gedistribueerde slimme *morphing* vleugelconcepten, hun rijping en acceptatie

in nieuwe vliegtuigontwerpen.

In de vorige delen werd het concept van de actieve *morphing* vleugel demonstrateur voorgesteld in de vorige paragrafen, en de haalbaarheid ervan werd onderzocht door middel van incrementele verbeteringen en opeenvolgende windtunneltests. Een belangrijke bevinding was dat de bandbreedte van de actuator en de kwaliteit van de ruisonderdrukking van de sensor kritische factoren waren die nodig waren om de grenzen te verleggen voor multi-objectieve optimalisatie en beperking van aero-elastische belastingen. Het laatste deel onderzoekt de impact van een snellere activering op de prestaties van de belastingvermindering en het vermogen om snellere aero-elastische doelstellingen te bereiken. Voor dit doel werd een gedistribueerde over-geactiveerde aero-elastische vleugel demonstrateur ontwikkeld, de SmartX-Neo, die een snellere activering mogelijk maakt met een veel eenvoudiger klepmechanisme en dezelfde geïntegreerde detectiecapaciteit garandeert als de SmartX-Alpha.

Het effect van het ontwerp van de actuator werd bestudeerd door middel van een reeks windvlagensimulaties van *closed-loop* controle van een parametrisch aeroservo-elastisch model van de SmartX-Neo, voor windstoten met verschillende frequenties. Actuatordynamica werd in het systeem geïmplementeerd door de stijfheid en demping van een tweedeordesysteem te schalen. Met de nominale *Linear Quadratic Regulator (LQR)* regelaar werden reducties van piekbelasting tot 78 % bereikt in vergelijking met het geval met open lus. Er werd waargenomen dat een lagere schaal die overeenkomt met een snellere activering, aanzienlijke verbeteringen oplevert tot 11 % ten opzichte van de nominale configuratie van de actuator, wat leidt tot een reductie van windvlagen tot 84 %. Bovendien werd waargenomen dat de effectiviteit van snellere actuatoren verbetert voor windstoten met een hogere frequentie, wat het potentieel van SmartX-Neo bevestigt om effectiever om te gaan met snellere regeldoelen.

List of Publications

Each chapter in this dissertation is based on the following list of peer-reviewed scientific journals or conference publications. Publications without a Chapter number are not included but are dissertation related. Journal and conference publications which are non-dissertation related are listed on page xxiii.

Journal Publications

11. **T. Mkhoyan**, V. L. Stuber, X. Wang, I. Mkhoyan, R. De Breuker, and S. van der Zwaag, "Bio-inspired Distributed and Decentralized Real-time Control and Sensing Architecture for a Seamless Active Morphing Wing," *Engineering Journal*, 2022, to be submitted (Chapter II.1, Chapter III.1 and Appendix III.A1).

10. **T. Mkhoyan**, X. Wang, and R. De Breuker, "Aeroelastic Wing Demonstrator with a Distributed and Decentralized Control Architecture," *AIAA Journal*, 2022, submitted (Chapter IV.1).

9. **T. Mkhoyan**, O. Ruland, R. De Breuker, and X. Wang, "On-line Black-box Aerodynamic Performance Optimization for a Morphing Wing with Distributed Sensing and Control," *IEEE Transactions on Control Systems Technology*, 2021, accepted (Chapter III.3).

8. **T. Mkhoyan**, N. R. Thakrar, R. De Breuker, and J. Sodja, "Morphing Wing Design Using Integrated and Distributed Trailing Edge Morphing," *Smart Materials and Structures Journal*, 2022, under review (Chapter II.1).

7. R. De Breuker, **T. Mkhoyan**, N. Nazeer, V. L. Stuber, R. M. Groves, S. van der Zwaag, and J. Sodja, "Overview of the SmartX wing technology integrator," *Actuators Journal*, 2022, under review.

6. B. Sun, **T. Mkhoyan**, E.-J. Van Kampen, R. De Breuker, and X. Wang, "Vision-Based Nonlinear Incremental Control for A Morphing Wing with Mechanical Imperfections," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–13, 2022 (Chapter III.2).

5. V. L. Stuber, **T. Mkhoyan**, R. De Breuker, and S. van der Zwaag, "In-situ boundary layer transition detection on multi-segmental (a) synchronous morphing wings," *Measurement: Sensors*, vol. 19, p. 100356, 2022.

4. X. Wang, **T. Mkhoyan**, and R. De Breuker, "Nonlinear Incremental Control for Flexible Aircraft Trajectory Tracking and Load Alleviation," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 1, pp. 39–57, 2022.

3. O. Ruland, **T. Mkhoyan**, R. De Breuker, and X. Wang, "Black-box Online Aerodynamic Performance Optimization for a Seamless Wing with Distributed Morphing," *Journal of Guidance, Control, and Dynamics*, p. 1, 2021, under review (Chapter II.2).

2. X. Wang, **T. Mkhoyan**, I. Mkhoyan, and R. De Breuker, "Seamless Active Morphing Wing Simultaneous Gust and Maneuver Load Alleviation," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 9, pp. 1649–1662, 2021 (Chapter I.1).

1. **T. Mkhoyan**, C. C. de Visser, and R. De Breuker, "Adaptive Real-Time Clustering Method for Dynamic Visual Tracking of Very Flexible Wings," *Journal of Aerospace Information Systems*, vol. 18, pp. 58–79, jan 2021 (Chapter I.1).

Conference Publications

10. **T. Mkhoyan**, X. Wang, and R. De Breuker, "Distributed Over-actuated Aeroelastic Wing Demonstrator with a Decentralized Control Architecture," *AIAA Scitech 2022 Forum*, 2021 (Chapter IV.1)[2].

9. O. Ruland, **T. Mkhoyan**, R. De Breuker, and X. Wang, "Black-box Online Aerodynamic Performance Optimization for a Seamless Wing with Distributed Morphing," *AIAA Scitech 2022 Forum*, p. 1840, 2022 (Chapter II.2).

8. **T. Mkhoyan**, R. N. Thakrar, R. De Breuker, and J. Sodja, "Design of a Smart Morphing Wing Using Integrated and Distributed Trailing-Edge Camber Morphing," in *Proceedings of the ASME 2020 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, (Irvine, USA), 2021 (Chapter II.1)[3].

7. **T. Mkhoyan**, C. C. de Visser, and R. De Breuker, "Adaptive state estimation and real-time tracking of aeroelastic wings with augmented kalman filter and kernelized correlation filter," in *AIAA Scitech 2021 Forum*, p. 0666, 2021 (Chapter I.2).

6. **T. Mkhoyan**, N. R. Thakrar, R. De Breuker, and J. Sodja, "Design and Development of a Seamless Smart Morphing Wing Using Distributed Trailing Edge Camber Morphing for Active Control," in *AIAA Scitech 2021 Forum*, p. 0477, 2021 (Chapter II.1).

5. A. R. R. Fassah, **T. Mkhoyan**, and C. C. de Visser, "Developing the Model Reduction Framework in High Frame Rate Visual Tracking Environment," in *AIAA Scitech 2021 Forum*, p. 0663, 2021.

4. **T. Mkhoyan**, C. C. de Visser, and R. De Breuker, "Parallel Real-Time Tracking and 3D Reconstruction with TBB for Intelligent Control and Smart Sensing Framework," in *AIAA Scitech 2020 Forum*, p. 2252, 2020 (Chapter I.2 and Chapter III.3).

3. **T. Mkhoyan**, C. C. de Visser, and R. De Breuker, "Adaptive Real-Time Clustering Method for Dynamic Visual Tracking of Very Flexible Wings," in *AIAA Scitech 2020 Forum*, p. 2250, American Institute of Aeronautics and Astronautics, 2020 (Chapter I.2 and Chapter III.3).

2. **T. Mkhoyan**, C. C. de Visser, and R. De Breuker, "Fast Symmetry Detection with Deep Learning and GeConv," in *IEEE RAS 2019 International Summer School on Deep Learning for Robot Vision*, 2019 (Chapter I.3).

1. **T. Mkhoyan**, "Adaptive state estimation and real-time tracking for control of a flexible wing with machine learning and AI," in *Global Innovation Forum 2019: Transforming Intelligence*, 2019 (Chapter I.3).

[2]Awarded *the Jefferson Goblet* at AIAA Scitech 2022 Conference, San Diego, CA, USA
[3]Awarded *Best student hardware* and (Runner-up) *Best Student Paper* ASME SMASIS 2020 Conference, Irwine, CA, USA.

Non-Dissertation Related Publications

5. **T. Mkhoyan**, M. Wentink, B. de Graaf, M. Mulder, *et al.*, "Mitigating Coriolis Effects in Centrifuge Simulators Through Allowing Small, Unperceived G-Vector Misalignments," *arXiv preprint arXiv:2202.02677*, 2022

4. **T. Mkhoyan**, "Investigation of the Coriolis effect in rotating space platforms," *ROOM Space Journal of Asgardia*, vol. 22, no. 4, pp. 24–29, 2019

3. **T. Mkhoyan**, "Investigation of The Coriolis Effect in Rotating Space Platforms for Space travel," in *Asgardia Space Science & Investment Congress (ASIC) 2019*, 2019

2. **T. Mkhoyan**, M. Wentink, M. M. van Paassen, M. Mulder, and B. de Graaf, "Mitigating the Coriolis Effect in Human Centrifuges by coherent G-misalignment," in *AIAA Scitech 2019 Forum*, p. 0714, 2019

1. V. Papadakis, M. P. de Boer, **T. Mkhoyan**, B. van Velzen, K. Seymour, and R. Groves, "Spectral imaging of Dutch gilt leather for improved conservation strategies," in *11th conference on Lasers in the Conservation of Artworks: Book of abstracts*, pp. 44–44, 2016

1

Introduction

Ex nihilo nihil fit.
Nothing comes out of nothing.

Lucretius, 60 BC, adapted from Parmenides

1

Advancements in aerospace engineering, paired with continuing desire to develop more fuel-efficient aircraft, have led to increasingly flexible aircraft designs. Generally, flexibility is considered a side effect of the lighter aircraft design and needs to be adequately accounted for to prevent undesired aerodynamics-structure couplings and ensure the optimised aerodynamic shape. While the flexibility can be accounted for with either passively tailored structural design or active control mechanisms, a fixed-wing shape - generally optimised for the cruise condition - cannot be fully optimised throughout the flight envelope due to conflicting requirements [1]. A more natural approach is to *utilise* the flexibility and actively change the shape by in-flight morphing. This allows the wing to continuously adapt to the most optimal shape when transitioning from one flight phase to the other. Secondly, compared to the conventional discrete trailing-edge surfaces, smooth morphing can execute flight control and load alleviation commands with reduced noise and drag. Collectively, these two aspects can contribute to a more efficient flight routine and a reduced structural weight, thereby improving flight sustainability.

Many research efforts have been devoted to morphing, including piezoelectricity, shape memory alloy materials, a compliant actuation mechanism, etc. [2]. Among all these research aspects, actuation force reduction is one of the bottlenecks of morphing realisations. Previtali et al. [3], reduced the actuation force by a compliant skin mechanism and a combination of conventional and piezoelectric actuation. However, this approach results in significant manufacturing challenges and complexities. Other concepts, such as the fishbone active camber (FishBAC) [4], and the mission adaptive digital composite aerostructure technologies (MADCAT) [5] demonstrate morphing with ultralight structures. However, most of the wing volume is consumed for morphing mechanisms, leaving little room for other components. Overviewing the state-of-the-art, the critical shortcomings of existing morphing techniques include 1) restricted morphing motions; 2) manufacturability and scalability complexities; 3) compromised internal wing volume; 4) inadmissibility for distributed morphing control along the wing span.

This dissertation proposes a distributed seamless active morphing wing concept to overcome these shortcomings. The morphing wing concept named the SmartX-Alpha is realised through the development of all aspects of the smart morphing wing, which encompass smart sensing, control and integration of decentralised real-time control system architecture. A second wing demonstrator is developed, the SmartX-Neo, implementing the learnings from SmartX-Alpha, to accelerate the adoption and improvement of distributed over-actuated wing concepts into new aircraft concepts. With this, the dissertation sets a critical step forward towards a greener outlook for future aircraft concepts.

1.1. Active Morphing Building blocks

While the previous section presented the problem statement, the importance, and applications of a smart actively morphing system, it is now critical to take a step back and understand the main components of such a system. Imagining an empty canvas, one can think of several building blocks which can be used to construct an actively morphing *system*. Firstly, a *model* or *data* is required to represent the system's behaviour. Secondly, the system must react intelligently through a *control* system. Much like in nature or any system in real-life, its output cannot be interpreted directly. Hence a *sensor* system is required to perceive the environment and dictate the behaviour of the morphing system.

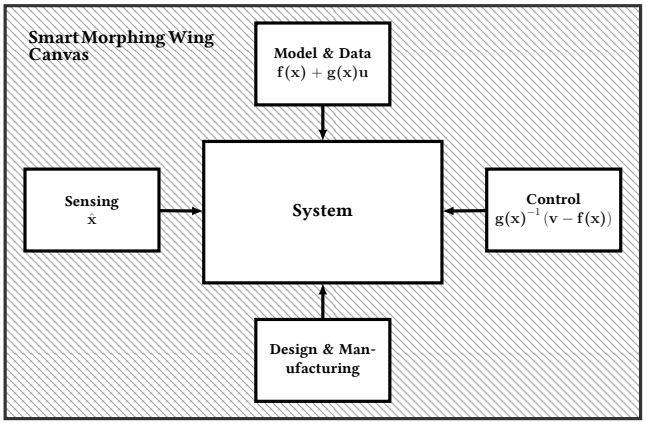


Figure 1.1: Building blocks of an active morphing wing system.

Finally, a suitable design and manufacturing approach must be chosen to realise the system physically. This analogy is adopted to represent the building blocks of the actively morphing system considered in this dissertation, illustrated in Fig. 1.1.

The connection of these blocks can be illustrated through a simplified control diagram. As shown in Fig. 1.2, the intelligent controller actively controls the system. The true system is observed utilising sensor measurements and reconstructed using prior data or output of the model. The estimation is fed back to the controller, closing the loop.

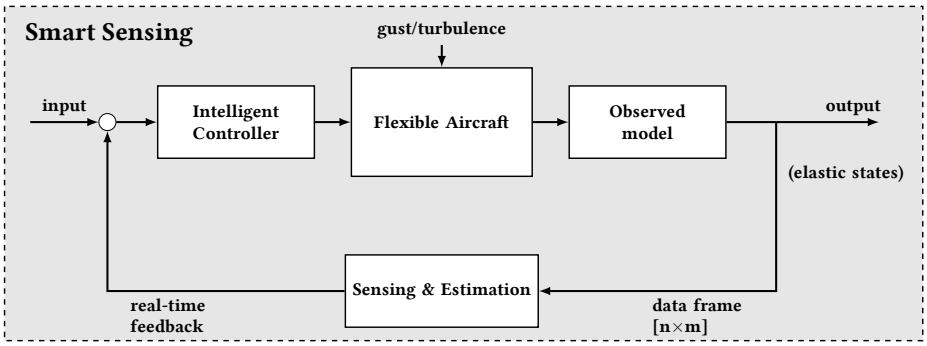


Figure 1.2: Simplified control diagram illustrating the interconnection of the morphing systems' building blocks.

1.2. Research Goals and Objectives

It is evident that, in order to arrive at an optimal active morphing system, in terms of design and performance, a *smart and multidisciplinary* approach is required that addresses each building block in the early design phase and integrates a multi-objective and robust optimisation framework considering all these components.

1.2.1. This Dissertation

The problem must be viewed holistically for such a smart system to be realised and optimally impact future, greener aircraft concepts. The goal and focus of this study are, therefore, to:

Research Goal

Develop the smart morphing framework for real-time in-flight performance optimisation, through multidisciplinary parallel integration of sensor-based control laws and the smart distributed sensing and actuation systems.

Advancing the goal of the study requires the state-of-the-art to be extended in a multidisciplinary manner by addressing the following central research question:

Research Question

How can multidisciplinary integration of sensor-based control laws, model-free sensing methods, and actuation mechanisms be used for real-time, in-flight, multi-objective optimisation framework of actively morphing wings?

1.2.2. Research Questions Breakdown

Falling back to the building blocks presented in Figure 1.1, the modelling, control, and integration of sensors and actuators aspects must be addressed accordingly to arrive at an integrated smart active morphing wing concept and advance the state-of-the-art. Therefore, the main research question is augmented with the sub-questions presented below. These questions are answered throughout this dissertation in the scope of (i) real-time performance optimisation, (ii) robustness and adaptability, which is the ability of the model and the control laws to overcome (adapt to) the model uncertainties (model and sensor errors), occurrences of system faults. The connection of the research questions is presented in Figure 1.3.

A ***Model & Data:*** *How can a fully coupled aeroelastic model of a wing be integrated into a real-time onboard multi-objective framework without compromising accuracy, and how can this model exhibit self-learning through the feedback of smart sensors?*

B ***Control:*** *Is it possible to reduce the model dependency of current control allocation methods through the use of sensor-driven, distributed actuation methods?*

C ***Sensing:*** *Can the multidisciplinary use of smart sensors and actuator systems, integrated with distributed, sensor-driven control designs, allow active shape control for improved aerodynamic performance and load alleviation of morphing wing systems?*

D ***Design and Manufacturing:*** *How can the controller, model, and sensing methods be integrated efficiently and holistically into a manufacturable morphing wing design capable of utilising distributed smart sensing and actuation mechanism without compromising the control and aerodynamic performance?*

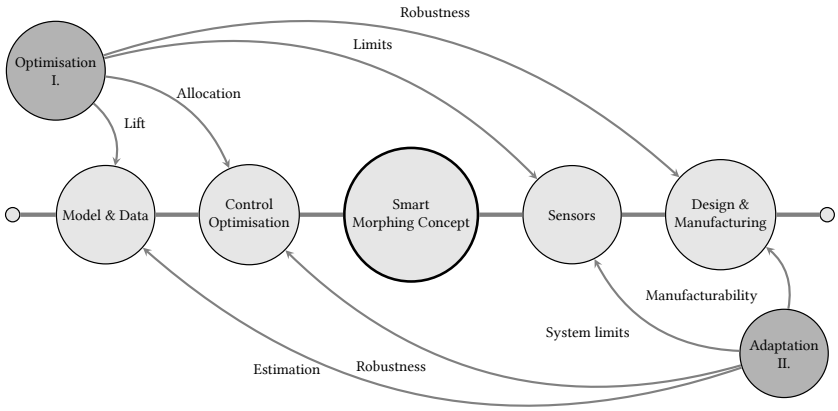


Figure 1.3: Rationale behind the research and the required topics to be investigated in the literature.

1.3. Dissertation Scope and Projects

The dissertation is conducted within the scope of the SmartX demonstrator project and has the following goal: **to develop and test a fully autonomous morphing wing**, capable of (i) optimal aerodynamic performance, (ii) desired flight dynamic behaviour, (iii) minimal loads to achieve minimal structural weight and (iv) automated high lift. The SmartX wing demonstrator must demonstrate autonomous morphing and performance optimisation capability with a distributed array of trailing edge flaps[1] in a Wind Tunnel (WT).

1.3.1. Project Structure

Considering the multidisciplinary nature of the project, the secondary aim of the SmartX demonstrator is to facilitate collaboration between the Aeroelasticity group Aerospace Structures and Computational Mechanics (ASCM), Non-Destructive Testing (NDT), Novel Aerospace Materials (NovAM) and the Control and Simulation (C&S) groups and act as a test-bed for integration of state-of-the-art developments of sensors, materials and actuators within these groups. An overview of the project structure and the focus of this dissertation within the project is illustrated in Fig. 1.4. The blocks SmartX-Neo/Alpha represent the demonstrators that have been developed in this dissertation, independent (SmartX-Neo) and in collaboration with others (SmartX-Alpha).

1.3.2. Control Objectives

The goals and the investigation outlined in this dissertation manifested in two wing demonstrators, the SmartX-Alpha and SmartX-Neo. The purpose of these demonstrators was to tackle several real-time objectives: performance optimisation of multiple objectives such as (i) drag optimisation, (ii) load alleviation, (iii) aeroservoelastic control and (iv) shape

[1]Distributed system of control surfaces (input effectors) categorises the wing as an over-actuated wing, i.e. more effectors are available than to be controlled variables (Degrees-of-Freedom (DOF))

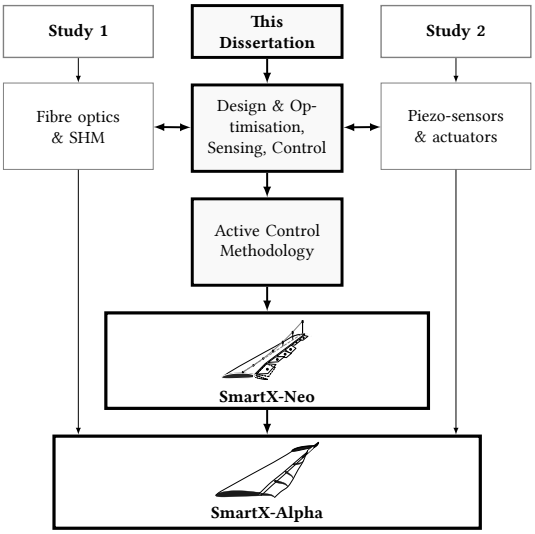


Figure 1.4: Overview of the Work Breakdown Structure of the SmartX project and the role of this dissertation.

control through multidisciplinary utilisation of smart sensing, control, actuation, and integration. The scope of the objectives covered by each wing demonstrator is presented in Fig. 1.5. The difference between the two demonstrators is the capability to morph continuously and smoothly (SmartX-Alpha), versus a faster, simpler actuation mechanism, allowing to address faster objectives (SmartX-Neo).

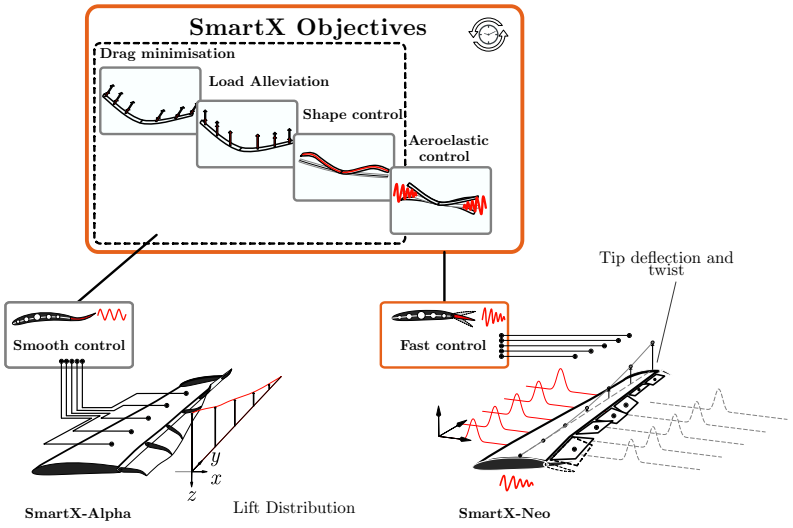


Figure 1.5: Comparison of the objectives of the SmartX-Alpha and SmartX-Neo.

1.4. Research Contributions

The contributions of this dissertation encompass the complete holistic process of the design, realisation, validation and improvement of an innovative smart morphing wing concept. The contributions are made in morphing design, analysis, sensing and control.

- Development of a novel vision-based sensing approach for morphing and flexible aeroelastic aircraft concepts, suitable for real-time implementation in a multi-objective control scheme in a non-invasive manner, presented in Part I. A novel unsupervised clustering method, presented in Chapter I.1, which is expanded with the complete shape reconstruction method in Chapter I.2. Finally, the implementation of deep learning with raw pixel data utilised in the sensing process presented in Chapter I.3, sets forth a new angle to vision-based sensing for flexible structures.

- Design, development, realisation and experimental testing of a distributed modular morphing concept, SmartX-Alpha, presented in Part II, Chapter II.1. The concept and system are realised in software and hardware through the development of a novel decentralised distributed control system architecture, validated experimentally through independent wind tunnel tests, presented in Chapters III.1, III.2 and III.3. In Chapter II.2, a novel learning-based shape-optimisation approach is developed for this morphing concept, for optimising steady-state lift-to-drag ratio, and evaluated numerically.

- Development, implementation and experimental validation of novel incremental control allocation method, integrated into a novel control system architecture allowing efficient and real-time processing of distributed morphing wing system presented in Part III. In Chapter III.1 simultaneous manoeuvre and load alleviation are demonstrated with an incremental control approach, ensuring relative actuator constraints and smooth allocation of the morphing system.

- Improving the performance of the morphing concept by addressing the actuation nonlinearities associated with the backlash hysteresis with a vision-based nonlinear incremental control method, introduced in Chapter III.2. In Chapter III.3 the shape optimisation objective is addressed, where the novel learning-based aerodynamic performance optimisation approach is demonstrated in a complex experimental system connected to various critical components of the morphing wing concept. The method demonstrated that online real-time performance optimisation can be achieved globally with an adaptive onboard model.

- An important step is made in Part IV with the development and evaluation of a second wing demonstrator, the SmartX-Neo, towards technology maturing and adoption of over-actuated distributed smart morphing wing concepts in new aircraft designs. The study, presented in Chapter IV.1, addresses how faster actuation and a simple actuator mechanism can improve load alleviation and the final objective, the aeroelastic control.

1

1.5. Reader's Manual

This dissertation consists of four Parts aiming to address (i) sensing, (ii) design, and (iii) control and integration aspects of a smart morphing wing concept. The final Part is presented as a (iv) future outlook, to help accelerate the adoption of morphing and distributed wing technology into new aircraft concepts by implementing the learnings from previous Chapters. This Part also contains the conclusion. Each Chapter is written with the aim to be published in a peer-reviewed scientific journal or has already been published and can be read entirely independently. Therefore, *the content for most Chapters is nearly identical to the published papers.*

A full list of publications related and non-related to this dissertation is presented at the beginning of this dissertation. To aid the reader in understanding various components and their interaction addressed in this dissertation, a visual guide is presented in Figure 1.6. The outline of the thesis is presented below.

The state-of-the-art review concerning the components above of the morphing wing is presented in Chapter 2. Part I focuses on developing a novel vision-based sensing approach for morphing and flexible aeroelastic aircraft wings, suitable for real-time implementation in a multi-objective control scheme in a non-invasive manner. The three Chapters presented in this Part are centred around the first wind tunnel test, *WT I*, conducted with an aeroelastic wing and a non-invasive camera tracking system developed for the study in the Open Jet Facility (OJF). In Chapter I.1, a novel unsupervised clustering method is presented using a machine learning approach. This is followed by a full shape reconstruction method in Chapter I.2, utilising Augmented Extended Kalman Filter (AEKF) for retaining the dynamic model of the wing subject to oscillations induced by gust vanes. Chapter I.3 presents a new direction toward vision-based sensing by utilising a Deep Convolutional Neural Network (DCNN) to extract information from raw-image data instead of the traditional computer vision approach. A comparison is made between traditional and Deep Learning (DL) based approaches, and recommendations are made on replacing the complete reconstruction process by DL.

In Part II, the design, development, realisation and experimental testing of a distributed modular morphing concept, SmartX-Alpha, is presented. In Chapter II.1 the morphing wing concept, the analysis methods centred around an efficient Fluid-Structure Interaction (FSI) framework, and the manufacturing methods facilitating the realisation of the morphing wing hardware demonstrator are presented. The morphing shapes are assessed, and mechanical imperfections, contributed mostly by backlash hysteresis, are characterised in experimental validation with Digital Image Correlation (DIC). This effect is later addressed by a compensating vision-based control approach in Chapter III.2. The concept and system are realised in software and hardware through the development of a novel decentralised distributed control system architecture, validated experimentally in independent wind tunnel tests in presented Chapters III.1, III.2 and III.3. A novel learning-based shape optimisation approach and a numerical evaluation of the distributed morphing wing concept are presented in Chapter II.2. This Chapter investigates the benefits of distributed morphing design in the context of online shape optimisation. It proposes an evolutionary-based optimisation strategy coupled with a Radial Basis Function (RBF) Neural Network (NN) to optimise the steady-state lift-to-drag ratio for a given target lift coefficient.

Following up on the previous Part, the focus of Part III is the control aspect of the morphing system, tested experimentally in two independent wind tunnel experiments, *WT II* and *WT III*. First, a sensor-driven, incremental control method is proposed for simultaneous gust and manoeuvre load alleviation based on Incremental Non-linear Dynamic Inversion (INDI) and Quadratic Programming (QP), demonstrating enhanced robustness to model uncertainties and backlash. The control approach is paired with an integrated distributed control system architecture allowing efficient and real-time processing of distributed morphing wing system in Chapter III.1. Using this control approach, simultaneous manoeuvre and load alleviation are achieved while ensuring relative actuator constraints and smooth allocation of the morphing system in the OJF wind tunnel equipped with a gust generator. Following this, a complex experimental system is a setup where all previous elements of an active morphing wing are combined to investigate online performance optimisation. In Chapter III.2, the performance of the morphing concept is improved by addressing the actuation nonlinearities associated with the backlash hysteresis effect. The vision-based nonlinear incremental control method is utilised to mitigate the effect in the wind tunnel setup effectively. Finally, Chapter III.3 addresses the shape optimisation objective, demonstrating a novel learning-based aerodynamic performance optimisation approach integrated into a complex real-time experimental system. Various key components of the morphing wing concept are validated in this final wind tunnel experiment in OJF. The method demonstrated that online real-time performance optimisation can be achieved globally with an adaptive onboard model.

The final Part, Part IV, is focused on the outlook of over-actuated distributed smart morphing wing concepts, their maturing and adoption in new aircraft designs. The last Chapter, Chapter IV.1, investigates the impact of faster actuation on the load alleviation performance and the ability to achieve faster aeroelastic objectives. For this purpose, a distributed over-actuated aeroelastic wing demonstrator was developed, the SmartX-Neo, allowing faster actuation with a much simpler flap mechanism and ensuring the same integrated sensing capability as the SmartX-Alpha.

Finally, the general conclusion and the recommendations are presented in Chapter IV.2, addressing the results of each study conducted to forward the development of the smart morphing wing technology.

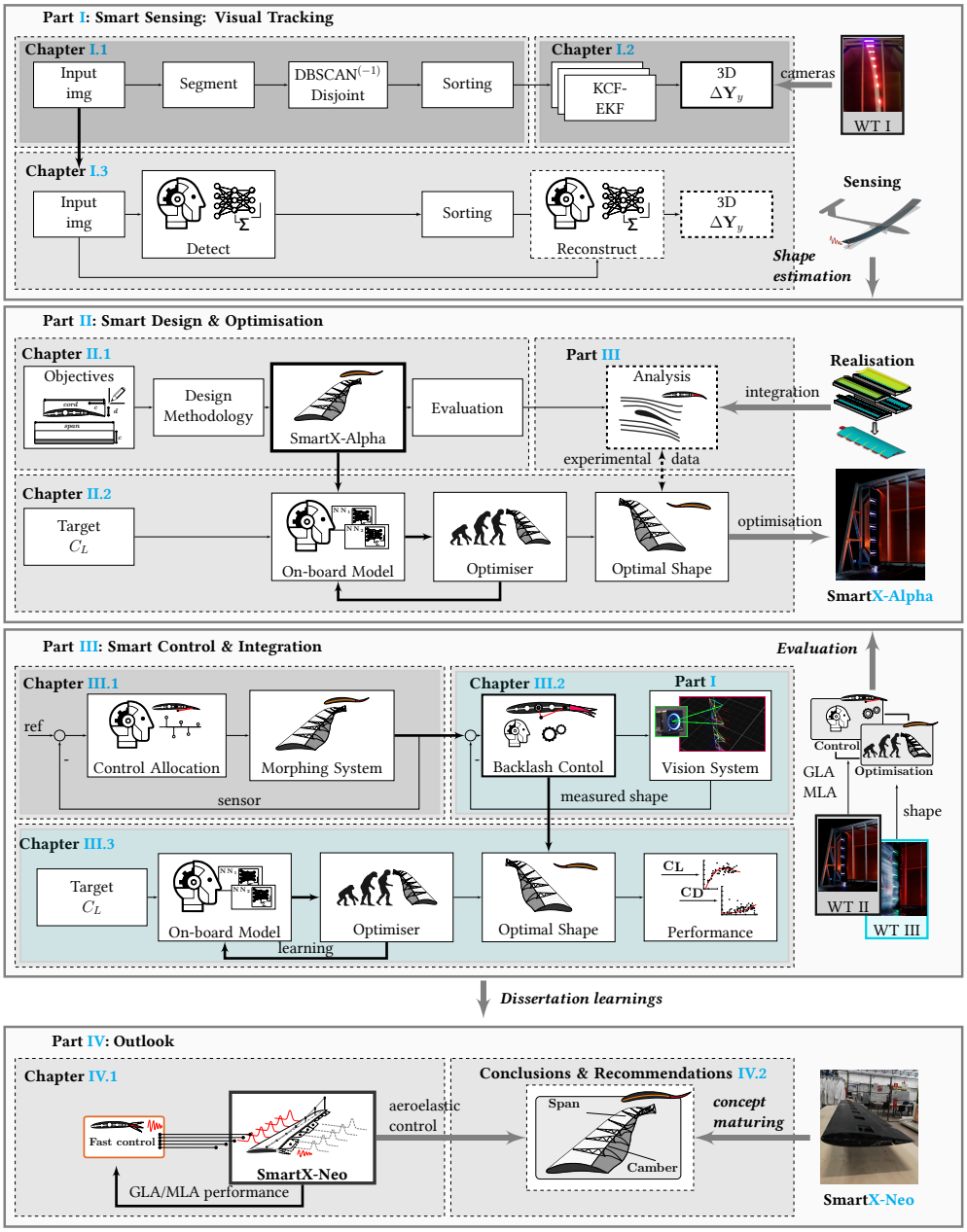


Figure 1.6: Dissertation outline.

References

[1] T. A. Weisshaar, "Morphing Aircraft Systems: Historical Perspectives and Future Challenges," *Journal of Aircraft*, vol. 50, pp. 337–353, Mar. 2013.

[2] J. E. Hubbard, "Dynamic Shape Control of a Morphing Airfoil Using Spatially Distributed Transducers," *Journal of Guidance, Control, and Dynamics*, vol. 29, pp. 612–616, May 2006.

[3] F. Previtali, A. F. A. Arrieta, and P. Ermanni, "Performance of a Three-Dimensional Morphing Wing and Comparison with a Conventional Wing," *AIAA Journal*, vol. 52, pp. 2101–2113, Oct. 2014.

[4] B. K. S. Woods, I. Dayyani, and M. I. Friswell, "Fluid/Structure-Interaction Analysis of the Fish-Bone-Active-Camber Morphing Concept," *Journal of Aircraft*, vol. 52, pp. 307–319, Jan. 2015.

[5] N. B. Cramer, D. W. Cellucci, O. B. Formoso, C. E. Gregg, B. E. Jenett, J. H. Kim, M. Lendraitis, S. S. Swei, G. T. Trinh, K. V. Trinh, and K. C. Cheung, "Elastic shape morphing of ultralight structures by programmable assembly," *Smart Materials and Structures*, vol. 28, p. 055006, May 2019.

2

Recent Advancements in Active Morphing

It is easy to explain how a rocket works
but explaining how a wing works
takes a rocket scientist.

Philippe Spalart, Boeing fellow

In the past decades, substantial research has been carried out in the area of active morphing. This chapter presents an overview of the literature on active morphing systems and gaps in the current state-of-the-art. The subsequent sections follow the structure of the building blocks of the morphing system presented in Fig. 1.1.

2

Over the past decades, substantial research has been carried out in the area of active morphing. This research encompasses not only the integrated morphing system as a whole but also advancements made to the individual elements, as illustrated in Fig. 1.1, out of which an active morphing system is composed—understanding the benefits and shortcomings of the current state-of-the-art active morphing systems and how the building blocks are connected is essential to advance the research further. The following sections present a literature review carried out in an attempt to gather insight into the state-of-the-art and current gaps.

2.1. Morphing and Design

In recent years, the technological advancements in aerospace materials, manufacturing technology, controller and hardware design allowed the development of increasingly lighter and more complex concepts. As the demand for sustainability and fuel efficiency continues to grow, the need for concepts which can increase flight performance becomes more evident. A particular technology which can significantly improve flight performance is active morphing. Initially inspired by avian biology, the effectiveness of such concepts is demonstrated in nature with wing shape adaptation and optimal gliding performance [1, 2].

As in nature, morphing wing concepts have evolved since the early years of aviation. One of the well-documented examples was the active roll control of the Wright Flyer, the first successful heavier-than-air powered aircraft. In this lightweight design, the lateral stability was ensured by wing twist-warping [3]. In this case, the flexible fabric-wrapped structure was well suited for morphing. However, as the flight speeds and loads increased with the advancement of aircraft design, a stiffer wing was required to fulfil load requirements and overcome aeroelastic instabilities. As a result, the conventional rigid wing design, generally optimised for the cruise, exhibits compromised performance in other flight conditions. More importantly, due to continuous fuel burn and redistribution of the weight, no optimal configuration can be found which is met through the entire cruise phase. Active morphing has the potential to reduce this performance gap and continuously optimise the aircraft performance across the entire flight envelope adaptively. However, a challenging aspect of active morphing is designing a feasible and effective morphing mechanism such that the aircraft performance can be improved actively throughout the flight envelope [4]. That is the subject of this dissertation.

In literature, various morphing concepts can be found. A comprehensive review of the early morphing concepts of various approaches regarding the actuator material, the actuation mechanism, and the skin types is presented by Barbarino et al. [5]. Examples vary from conventional to compliant mechanisms and materials in the latter two categories. Also, various materials are investigated for the actuators, ranging from conventional to piezoelectric and shape memory alloys. In aircraft wings, morphing can be applied to the leading edge, trailing edge, or both.

Kintscher et al. and Sodja et al. investigate a seamless morphing droop nose concept for the leading edge, designed to match a given target shape with different materials used for the morphing skin such as glass-fibre pre-preg and aluminium [6, 7]. The concept by Sodja et al. utilises conventional actuation. Here, low actuation forces are achieved by maintaining the skin length constant during morphing, such that strains in the skin are

minimal. Several other concepts achieve low actuation force by utilising compliant skin and actuation mechanisms [8–10]. While promising, the studies highlight the importance of further research into manufacturing and up-scaling complex compliant designs since the manufacturing process of these complex shapes is still challenging.

Further, examples of the compliant mechanism and actuation are investigated. Previtali et al. used conventional actuators, and Molinari et al. used piezoelectric skin actuation [9–12]. Some studies use bio-inspired design, such as the FishBAC concept, designed to mimic the compliant skeletal frame of fish, developed at Swansea University [13–15]. Trailing edge mechanisms are also presented by FlexSys, which have been installed and undergone flight tests on a modified Gulfstream III business jet [9, 16, 17].

Recent studies also investigated the use of ultralight, lattice-based structural modules assembled in a modular adaptive structure using Carbon Fibre-Reinforced Polymers (CFRP) [18–20]. The advantage is that these materials can have the stiffness of a typical elastomer at the mass-density typical to aero-gel. Cremer et al. [19] demonstrate improved aerodynamic efficiency and roll control authority with spatially programmed elastic morphing shape of a $4.27\ m$ wingspan aircraft in the wind tunnel. Jennet et al. [18] present the digital morphing wing concept constructed from discrete lattice elements. This concept shows increased roll efficiency compared to a conventional wing by applying spanwise twisting deformation. While promising, the lattice-based modules occupy most of the internal space due to their programmable flexibility and lightweight. Therefore, additional consideration is needed to ensure the flexibility of the structure while reserving the room for fuel, batteries, and other components to be installed in the wing. The lattice-based concept, presented by Keidel et al. [20], suggests a potential structurally efficient approach through optimisation of the orientation and distribution of the CFRP rods. However, this concept needs additional consideration for larger wing structures and manufacturability aspects.

Another study developed under NASA Advanced Air Transport Technology investigated the multi-flap Variable Continuous Camber Trailing Edge Morphing (VCCTEF) concept for the Generic Transport Model (GTM) [21]. This concept demonstrated effectiveness in multi-objective control and gust load alleviation in [22, 23]. However, the real-life experimental demonstrator with SMA rotary actuators revealed many challenges such as weight effectiveness, speed and power requirements of the actuators, the complexity of the multi-segment camber mechanism, and skin flexibility required [21].

In a recent EU FP7 CHANGE project study, a morphing concept called the Translation Induced Camber (TRIC) is introduced to address some of these problems [24]. This concept implements a relatively simple and effective morphing mechanism that uses a combination of cross-sectional warping and skin bending to induce camber and twist morphing with a pair of conventional actuators. The advantage of this concept is its relative simplicity and compactness of the actuation mechanism, which increases the fuel carrying capability and volume needed for necessary auxiliary components in the wing. However, the main disadvantage of the currently proposed TRIC concept is that the lift distribution cannot be influenced locally with a single morphing surface controlled by a single pair of actuators. As a result, this inhibits the use of the morphing mechanism for multi-objective flight control and limits its use as a direct replacement of conventional control surfaces for rigid body motion control (ailerons, rudders and elevators). Various control design studies

highlight the necessity and effectiveness of multi-objective flight control, load alleviation, and drag reduction performed by distributed multi-flap systems such as the VCCTEF investigated in [22, 23] and conventionally flapped over-actuated aircraft models [25].

Key observations of the literature study on modelling and analysis aspects:

- Current morphing concepts suffer from the complexity of the actuation mechanism, which hinders effective implementation and scalability.
- For smooth local control of the lift distribution, a distributed (morphing) system is necessary.

2.2. Control

In the previous section, several real-time objectives of the active morphing system were outlined for this dissertation, shown in Fig. 1.5. Manoeuvre Load Alleviation (MLA) and Gust Load Alleviation (GLA) are two important objectives in this context. Conventional MLA generally requires extensive tuning and relies on some pre-designed control logic to trigger pre-selected wing control surfaces when the measured load exceeds a pre-defined threshold. Another approach, by Pereira et al. [26] uses a linear Model Predictive Controller (MPC) and a Linear Quadratic Regulator (LQR) to satisfy the load constraints at various critical stations. In another study [27], the nonlinear flexible aircraft model is linearised successively, and then MPC controllers are designed at every linearisation point. A Linear Quadratic Gaussian (LQG) control is designed for a SensorCraft vehicle GLA problem in [28]. Another experimental study in the Open Jet Facility (OJF) investigated the potential of load reduction with smart rotors using H_∞ loop shaping and Feed Forward control [29]. Besides, a wind tunnel experiment for alleviating the gust loads of a flexible wing with piezoelectric control is presented in [30]. The piezoelectric patches are actuated by a Proportional-Integral-Derivative (PID) controller using wing-tip linear acceleration measurements. In the study by Baldelli et al. [31], an aeroelastic morphing vehicle is controlled using Linear Parameter-Varying (LPV) and pole placement techniques.

In recent years, several studies have been conducted within the scope VCCTEF [23]. This distributed control surface layout concept achieves multi-objective flight control and manoeuvre load alleviation. The multi-objective control approach is suitable for problems where several, and possibly conflicting, performance objectives are involved, such as minimising drag and loads or maximising manoeuvrability [23, 32]. Two cost functions are used in the LQG control, one for rigid-body command tracking and another for elastic mode suppression and the wing root bending moment minimisation [23]. Simulation results in [23] revealed that the pitch rate tracking performance is degraded when enabling the MLA function, reducing the ability to achieve the target loads for command tracking. This contrasts with the belief that physically, an aircraft possessing the ability to deploy distributed control surfaces should be able to alleviate loads from manoeuvres and gusts while maintaining the required loads for command tracking. In a follow-up study [33], the LQG controller is combined with an adaptive GLA function, which estimates the gust components on rigid-body and elastic dynamics using an adaptive gradient law. However, the estimation is impaired due to the limited number of accelerometers, model uncertainties

and high learning rates for the adaptation laws required to estimate the high-frequency components of the disturbances.

An *active morphing wing* is a complex system that must adapt to constantly changing flight conditions and disturbances and can also exhibit coupled and nonlinear interactions between aerodynamics and structural dynamics. While conventional linear control methods, paired with the gain-scheduling and identification methods, can demonstrate adaptiveness, a tedious tuning process is required. Furthermore, with this approach, the resulting stability and performance cannot be guaranteed between operational points [34].

Moreover, due to the complexity of the actuation mechanism and manufacturing imperfections, nonlinearities can arise in the actuation behaviour of the morphing surfaces, which are also characterised in other aerospace systems as [35], friction [36], deadzone [37]. These nonlinearities can degrade the system performance and lead to undesirable phenomena such as limit-cycle oscillations, divergence, and even flutter [37, 38]. In particular, in the presence of coupled and nonlinear interactions between aerodynamics and structural dynamics, this imperfection can magnify the uncertainties with a complex wing system. Therefore, a sensor-driven nonlinear control approach is required, which is less model-dependent, can cope with these uncertainties and can be paired with an online identification routine to adapt to constantly changing flight conditions and disturbances.

Incremental Non-linear Dynamic Inversion (INDI) control law is a sensor-driven nonlinear control method which does not require gain scheduling and has also been used to alleviate loads effectively in [39]. Due to reliance on sensor data, this method can cope with model uncertainties. It is less model dependent compared to other model-based nonlinear control methods such as feedback linearisation [40] and backstepping [41], known for *full model knowledge assumption* [42]. This incremental sensor-driven method has also been extended to backstepping with Incremental Backstepping (IBKS)[43]. Experimental and simulation results have demonstrated the robustness of INDI to model uncertainties [44], gust disturbances [39], actuator faults [45], and structural damage [46]. There are several other benefits to INDI, such as low computational load, suitability for real-time implementation, and uncertainty parameterisation process instead of adaptive control methods.

The study by Wang et al. [39] demonstrated that a trade-off must be made for load alleviation with INDI when the flexible aircraft has a conventional aileron configuration. Consequently, the performance could be improved by utilising an over-actuated distributed flap configuration intended for SmartX-Alpha; however, a control allocation strategy must be implemented for such a system. In the literature, an Incremental Nonlinear Control Allocation (INCA) method has been proposed for a tailless aircraft with Innovative Control Effectors (ICE) [47]. While this approach provided effective control allocation for a highly manoeuvrable nonlinear aircraft model, the relative position constraints were not considered. In order to enforce smoothness of the lift distribution and allocation for a distributed morphing concept, relative position constraints and consequent smooth actuator allocation are critical.

Attempts to tackle computational efficiency of control allocation of large distributed systems have also been addressed by parallel computing in [48], for distributed wavefront reconstruction of large-scale adaptive optics system [48]. Aside from model or sensor-driven approaches, black box methods are currently an active area of research. Deep Reinforcement Learning (RL) has gained significant traction as a black-box, easy-to-implement

alternative in the control design of linear and nonlinear systems [49–51]. The promise of this method is the ability of the agent (controller) to learn an optimal policy through observation of the inputs and outputs of the system in a purely online and adaptive manner, without prior knowledge of the system. Studies are demonstrated for stabilisation of heave dynamics of a flexible wing-fuselage model [52], and online control of flexible aircraft using Integral RL in [53]. The latter variation of RL was developed to compensate for drift dynamics and facilitate the convergence to the optimal solution without explicit knowledge of the internal dynamics of the system [54]. While promising, the challenge for this method remains the lack of stability guarantee, the volatility towards the learning parameters, and the structure of the Neural Network (NN) used to capture the input-output data structure.

It seems that to fully benefit from the potential of advanced sensor-driven control methods, the control system architecture must mitigate the lag in the system and allow efficient sampling and data communication. Secondly, the sensor dependency must be mitigated to maximise the robustness against model uncertainties. This will be a key challenge to overcome for an over-actuated and over-sensed system envisioned in this dissertation.

Key observations of the literature study on control aspect:

- Computentionally efficient real-time control methods are desired, which can utilise sensory data available onboard of smart morphing system.
- Control allocation approach must be developed, which considers relative actuator constraints and ensures smooth allocation.

2.3. Sensing

As shown in the simplified control diagram in Fig. 1.2 the sensing aspect is tightly coupled with the control method. In the previous section, it was already highlighted that less model-dependent control methods are beneficial due to ease of implementation, efficiency and their ability to adapt to model uncertainties. INDI is one such control method that helps reduce model dependence; however, it relies more on sensory input to cope with model uncertainties. For these types of methods, the quality of the sensor and the estimation it provides are critical factors, as any sensor uncertainties would negatively impact the effectiveness of the controller.

With the rise of more flexible and unconventional (over-actuated) aircraft designs (such as VCCTEF and X-56A [55]) and sensor-dependent control methods, the need for alternative robust sensing methods increases; these are twofold. One is the need to have robust model-free sensors, such that the limitation of the *perfect sensor assumption* of incremental methods is tackled. Second, many parameters must be measurable and are required as state feedback for an aeroelastic system to properly account for multiple objectives such as gust loads, control effectiveness, flutter etc. A crucial aspect for the majority of these objectives is an accurate and efficient estimation of the wing shape and control surface deflections. To address these needs, new sensing methodologies and sensor hardware comprising conventional and non-conventional smart sensors must be considered.

The study by Weisshaar et al. [4] highlights the ability to monitor and communicate structural state information as one of the critical aspects of developing smart morphing structures. Conventional sensors such as accelerometers, gyros, optical fibres and strain gauges are widely used for shape estimation in flexible aircraft [33, 39, 56]. However, these conventional sensors suffer from temperature and gyroscope/sensor drift and provide low sampling rates (optical fibres). Furthermore, as indicated in [33], the estimation can be impaired by the choice of the sensor layout, which, due to the requirement to embed the sensors in the wing structure, makes this approach model-dependent and require high precision placement.

Visual sensing is a solution that can reduce the complexity associated with hardware installation and provide shape feedback in a non-invasive manner. This type of sensing can replace an array of conventional and allow observing multiple nodes of the system's flexible states using an image sequence from a set of fuselage-mounted cameras. The use of visual information for observing deformations has been successfully implemented on wind tunnel models in early studies [57] and has also seen widespread application in robot manipulation [58]. Various other vision-based applications for aerial systems have already been investigated in recent studies. Applications range from for areal imagery [59–61], to aerial navigation [62] and flight control tasks, such as aerial refuelling [63], landing [64] and estimation of rigid body aircraft states, such as altitude [65]. In a flight-test experiment conducted by the Netherlands Aerospace Centre (NLR), it was shown that a Digital Image Correlation (DIC) system was capable of providing accurate measurement of the displacements in the millimetre range [66]. However, a speckle pattern was required to ensure accurate measurement. The development of robust marker-based tracking methods can help reduce reliance on special coating [67].

Light Detection and Ranging (LIDAR) is another sensing technology which could benefit smart morphing wing technology by providing a feed-forward path to the controller for the upstream disturbances and the flow field. The potential of this system was studied, and a LIDAR based system, Molecular Optical Air Data System (MOADS), was proposed in [68]. Off-line flight tests were also performed with NLR for one-dimensional measurements of free-stream velocity [69], although limited flight tests with closed-loop control are performed for a full-scale two-dimensional flow field reconstruction. While gaining popularity in automotive applications, integrating this type of system in a small factor, low-cost computation unit is still challenging.

In recent years, the capability of onboard computation and the image processing quality has immensely increased, while the hardware has become more compact [70, 71]. The rise of low-profile, embedded, low-cost and high-performance Graphics Processing Unit (GPU) computing modules such as JetsonTX2 and Xavier [72] allow not only to process a collection of visual data but also enhance the computation by real-time hardware acceleration and onboard Artificial Intelligence (AI).

However, integrating vision-based sensing and other sensors for efficient real-time shape estimation for high degree-of-freedom systems such as distributed morphing is challenging. In particular, improving the robustness and sampling frequency of vision-based sensing is a crucial aspect that must be considered.

Key observations of the literature study on sensing aspect:

- Vision-based sensing is a promising, non-invasive alternative for model-free shape estimation.
- Improving the robustness and sampling frequency of visual-based sensing is a key aspect that must be considered when implementing this method.

2.4. Modelling, Analysis and Identification

Traditionally, aeroelastic modelling is motivated by passive design methodology. The motivation behind such an aeroelastic model is the ability to tailor the wing design through aeroelastic analysis while optimising for structural layout, material properties or, in the case of composite material, the directional stiffness of the material [73].

2.4.1. Aeroelastic Modelling and Analysis Frameworks

The aim of the aeroelastic modelling perspective has been the development of a design and optimisation framework, contributed by essential works of [73–76]. Recent works have also been conducted on extending the design space to full aircraft model and flight dynamics coupling [77]. In the work by Werter [76], the aeroelastic modelling approach had the aim to improve the conceptual design of aircraft wings through optimisation. The 3D wing geometry is generally represented by a discretised 1D structural beam model to improve the computational speed. The structural model can be extended for the static analysis with geometric nonlinearities, as implemented by Breuker [75]. The dynamic aeroelastic model is linearised around the static equilibrium [76] and coupled to the aerodynamic method of choice in the work of Werter [76]. Generally, two-dimensional unsteady airfoil theory, often referred to as trip theory, or aerodynamic panel methods are used, such as the Unsteady Vortex Lattice Method (UVLM) [78], in order to obtain a continuous state space representation of the system.

This state space representation can be used to evaluate the aeroelastic response of the system to disturbance input. Strip theory has been widely used for the analysis of High-Altitude Pseudo-Satellite (HAPS) [79, 80], applicable due to its computation efficiency and the ability to cope with lightweight, flexible HAPS platforms such as High Altitude Long Endurance (HALE), Zephyr [81, 82]. The state space system can be obtained through the finite-state method or Leishman's indicial method [83]. While actuator dynamics do not include damping terms, an aeroservoelastic model is obtained in [84] by augmenting the aeroelastic model with a lumped torsional spring representing the flap. Experimental validation of Stemme S15 aircraft conducted by Silvestre et al. [85] demonstrated that a suitable model for flight control law design could be obtained using strip theory and the indicial method.

Another widely used aerodynamic modelling method utilising the Aerodynamic Influence Coefficients (AIC)'s is Doublet Lattice Method (DLM) [86]. Its main advantage is the ability to capture compressibility; however, the model's accuracy can be impaired when converting from frequency domain to time domain using Rational Function Approximation (RFA) [76]. Given that the UVLM is formulated in the time domain, its advantage

is the ability to cope with flows around wings exhibiting large motion, as the transient response and the free wake are computers directly [87].

When higher fidelity is required, Computational Fluid Dynamics (CFD) based methods can be used for flutter [88] or gust response [89] analyses; however, the computational cost would render this approach unsuitable for control design purposes. Therefore, the cost must be reduced through model reduction techniques and a data-driven approach for approximating a CFD-based model. The main advantage of (U)VLM and strip theory models is the ability to obtain a low-cost coupled model in the time domain. However, while the UVLM model developed by Werter. [76] allows for the computation of the inviscid, incompressible, irrotational, unsteady aerodynamic forces and moments, the dynamic model is linearised around small perturbations around a steady-state reference configuration governed by a given aerodynamic shape and flow conditions. The model in this approach is typically analysis-driven for a given load case. Therefore, the model needs to be linearised and recomputed to cover multiple dynamic load cases and regions of the flight envelope.

It must be noted that to meet the real-time objectives, the smart morphing wing envisioned in this dissertation must be able to cope with varying dynamic load cases through its mission profile. Furthermore, linearisation obtained analysis framework presented in [76, 77, 90] can still be computationally infeasible in real-time for a high degree-of-freedom distributed control panel configuration such as the envisioned SmartX wing.

Another approach for obtaining a low-cost time-domain model is through model reduction techniques, where the purpose of the Reduced-Order Modelling (ROM) is to reduce the size of the model (i.e. number of states or modes) while retaining a sufficient level of accuracy. Several ROM approaches exist, ranging from Modal Truncation (MT), where the structural model is limited to a set of eigenvalues and corresponding eigenvectors [91], Balanced Truncation (BT) where the system is reduced based on its observability and controllability [92], Proper Orthogonal Decomposition (POD) or balanced POD obtained using singular value decomposition of the system at a given condition [93, 94]. In a recent comparative, MT showed the lowest computational cost but was impaired in accuracy compared to higher cost methods such as the BT [95]. Based on the efficient reduction of the controllability and observability, BT has effectively reduced a sizeable aerodynamic system composed of linearised Navier–Stokes equations [96]. While effective, this method is only suitable for linear systems. To fully account for a nonlinear aeroelastic model and retain the accuracy, potentially, the incremental variant of this method, suitable for nonlinear systems, was investigated in [97]. The reduced system can be further condensed when only a specific frequency range is selected for the desired application. The study by Gugercin et al. [98] addressed this aspect by proposing a frequency-weighted balanced reduction approach with an absolute error bound and stability guarantee.

2.4.2. Data-Driven Models and System Identification

Another approach that aligns with model reduction is a data-driven representation, or a surrogate model, of an aircraft model or a wing constructed from high-fidelity high-cost analysis such as CFD. A recent study proposed a CFD-based data-driven approximation of an aeroelastic aircraft model employing a Loewner-based data-driven optimal approximation and Eigensystem Realisation Algorithm (ERA) [99]. The use of a suitable surrogate model can also substantially improve optimisation performance in the context of

the Multi-disciplinary Design Optimisation (MDO) framework, as suggested by Alba et al. [100].

Experimental data gathered from the wind tunnel or in-flight experiments can also be used to validate or represent the model in an entirely data-driven manner. The model or the system can also be identified from data collected a priori or online from sensor measurements. Many system identifications and consequent control methods utilising the model rely on full-state feedback assumption. This is often not directly realisable but can be facilitated using sensor fusion techniques and Kalman Filter (KF) model, such as the case for LQG. One recent approach developed for data-driven identification is Dynamic Mode Decomposition with Control (DMDc), which aims to extract a low-order model suitable for real-time control from high-dimensional, complex systems [101]. Recent studies have demonstrated the development of data-driven nonlinear aeroelastic models for morphing wing control using an interpolation scheme to smoothly connected, linearised DMDc [102].

In summary, the literature study highlighted that aeroservoelastic models could be obtained through design-driven or data-driven models. For active control of an autonomous smart morphing wing, a shift is required from *design-driven* models for passive tailoring purposes to low-cost *active* models suitable for control purposes.

Key observations of the literature study on modelling and analysis aspect:

- A shift is required from *analysis models* for passive tailoring to *active* aeroservoelastic models for control purposes.
- Cost efficiency and accuracy are critical factors for establishing an aeroservoelastic and morphing wing model to be used both for control-driven purposes and for analysing many load cases.
- A distinction is made between design-driven models and data-driven, experimental models. The latter provides an alternative to unsteady aerodynamic models and is suitable for experimental identification and real-time control.

2.5. Gaps, Choices and Needs for this Dissertation

Several relevant aspects of the smart morphing system were outlined in the previous section. To advance the current state-of-the-art, the gaps in (i) morphing design, (ii) control, (iii) sensing and (iv) modelling and analysis must be filled.

2.5.1. Current Gaps and Choices

The following paragraphs describe the gaps and choices established concerning this dissertation's design, control, sensing, and modelling aspects.

Morphing Design It is evident that a distributed morphing concept must be developed to address the shortcomings of the previous morphing designs. Critical challenges of morphing concepts are the complexity of the actuation mechanism, the manufacturability, the scalability and the integration of design, control and sensing aspects. Therefore, to

maximise the impact of this dissertation and accelerate the adoption of smart morphing technology for future aircraft concepts, the effectiveness, manufacturability and scalability must be proven by a simple yet effective actuation mechanism while integrating all areas of the morphing system, as highlighted in Fig 1.1, through this dissertation.

Efficient Control Methods Smoothness and continuity of the distributed morphing system must be facilitated using adequate material or structural design and control allocation methods to ensure relative actuator constraints. Sensor-driven control methods, such as the INDI, are suitable because the morphing concept must facilitate various sensing capabilities. Furthermore, the model dependency must be reduced such that the control method is easy to implement and can deal with model uncertainties, which are expected from a complex system. Finally, a computationally efficient real-time control method is desired, utilising sensory data available onboard the smart morphing system to meet various objectives. Therefore, control allocation methods must also be efficient.

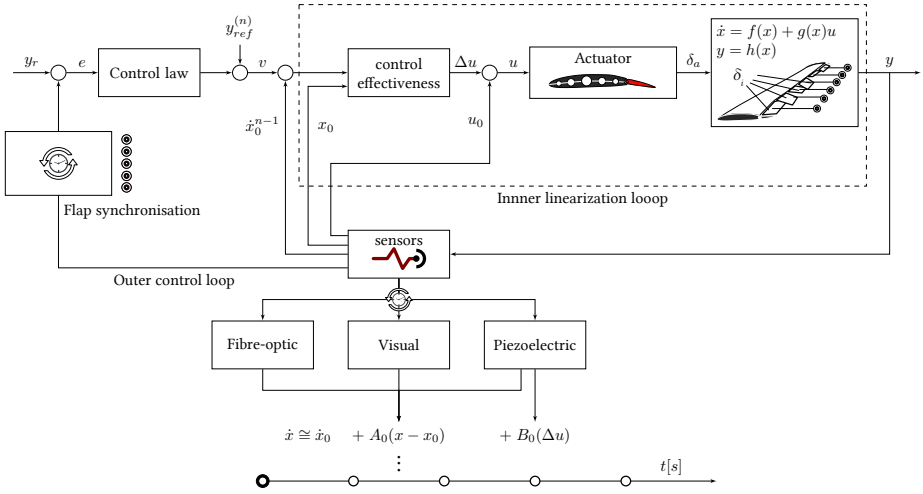


Figure 2.1: Simplified schematics of the proposed distributed control allocation and sensor synchronisation.

Robust Sensing and Data Fusion The morphing concept must facilitate various sensing capabilities to account for various real-time objectives. Shape sensing is the critical aspect required for most objectives (e.g. load alleviation, shape control, aeroelastic control). Vision-based sensing is a suitable method due to its model independency, ease of implementation and non-invasive nature. In the event that the configuration or the location of the desired nodes of the system must be adapted, a camera-based setup has a clear advantage in terms of flexibility and ease of implementation. Tracking and reconstruction algorithms must be developed suitable for efficient onboard computation. The robustness of the visual-tracking algorithms must be considered when implementing this approach. A possible enhancement of this sensing approach is incorporating dynamics-informed filters, such as the KF, into the reconstruction and tracking approach. The sensing robustness

can be further improved by implementing a sensor fusion strategy utilising multiple sensory data. The data and control system architecture will play a principal role in facilitating complex real-time sensing and control strategies. In Sec. 2.5.2 a potential approach is discussed. A simplified control schematic of the abovementioned components is presented in Fig. 2.1. In this scheme, the INDI based online controller linearises the plant using sensor feedback from novel sensors (optical, piezoelectric, fibre-optic).

Modelling, Analysis and Identification A key aspect in the design of the SmartX is the capability to autonomously implement multi-objective functions using the available sensor data onboard, which can be used in real-time in the wind tunnel or in-flight.

Implementing these multi-objective functions, either in-flight or in the wind tunnel, not only depend on sensor strategy but also on a coherent system identification methodology that aligns with control methods developed in parallel. Control methods that are adaptive in nature generally go hand in hand with well-chosen system identification strategies. It is more so the case for uncertain systems and/or undergoing experimental testing. In the context of experimental system identification, it serves, on the one hand, the purpose of building and correcting mathematical models from imperfect sensor observations. This allows identifying parameters needed for control law updates either in-flight or post-flight, reducing the model uncertainties. But also provides surrogate models with reduced order and sufficient fidelity for flight control law development and in-flight deployment. In this context, two categories of system identification can be performed, parametric and non-parametric. In the prior case, the identified model parameters represent the true physical parameters of the model, e.g. M_δ (pitching moment control effectiveness) and M_α (longitudinal stability derivative) [103]. In the latter category, the system's parameters are abstract and do not represent physical values. However, they can still relatively accurately represent the system behaviour and match the eigenvalues of the true system.

2.5.2. Need: Distributed Control System and Data Framework

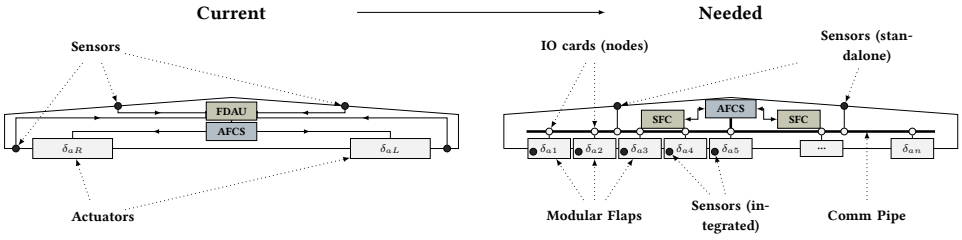


Figure 2.2: Need towards distributed real-time framework and synchronisation.

A principal aspect of a smart morphing system, which must utilise an abundance of sensors and actuators, is a control system architecture that can facilitate the integration of various hardware components and synchronise shared data from various sources. A conventional autonomous control system adopts strict data architecture, where the Flight Data Acquisition Unit (FDAU) and the Automatic Flight Control System (AFCS) are typically separated. In this non-modular configuration, the AFCS is responsible for the acqui-

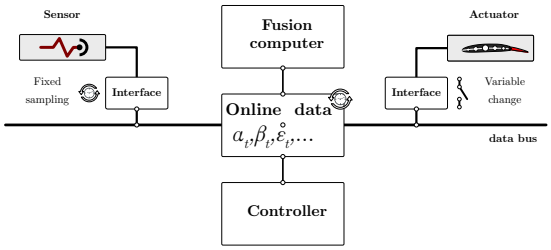


Figure 2.3: Real-time data synchronisation for sensing and control.

sition and processing of the sensors, the FDAU for generating the control signals, and the system is not scalable.

It is noteworthy to realise that for an over-actuated and over-sensed system to be realisable, a shift is required from the current conventional system to a modular and scalable approach. To meet this goal, decentralised real-time data architecture is envisioned, which would allow the distribution of the data processing effort across various processing nodes and the scalability of the system through modular design. The architecture shall be developed with adaptability in mind to allow the fusing of data from different hardware sources at different sampling rates, subject to the limitations of the sensing hardware and in-flight control function requirements. The differences between conventional and envisioned systems are illustrated in a simplified diagram shown in Fig. 2.3. In this approach, the sensory information (e.g., the strain measured at a node) and output information (e.g., actuator positions) can be represented in synchronised real-time variables with a decentralised communication bus topology. The principle of data synchronisation is illustrated in Fig. 2.3 The data architecture acts as a synchronising layer between physical hardware and the control software, updating parameters relevant to in-flight control functions. From the hardware perspective, the AFCS acts as the master synchronisation node, providing the timing and computation of the control laws. The Sensor Fusion Computer (SFC) is responsible for data fusion, onboard analysis and identification. Redundancy of the system can be ensured by adding additional processing or sensory units.

The envisioned configuration is particularly suitable for adopting AI capable, hardware-accelerated embedded computational units with GPU and Central Processing Unit (CPU) capability, such as the NVIDIA Jetson TX2 and the newer AGX Xavier. These low-profile (light and low power consumption) modules allow implementing native optimised Compute Unified Device Architecture (CUDA) libraries for solving AI or Machine Learning (ML) tasks online in a small form-factor (100x87, 20-40 W) with high performance: 32 TOPS (32 trillion operations per second); 750 Gbps high-speed IO; 512 CUDA cores, 64 tensor cores [72].

In summary, the benefit of the envisioned system includes: (i) the control functions have the flexibility to choose their inherent sampling rates; (ii) each sensor can be sampled at the optimal sampling rate of the sensor with its dedicated hardware (Analog to Digital Converter (ADC) converter etc.); (iii) controller tuning with Hardware-in-the-loop (HIL) becomes very flexible; (iv) ease of integration and up-scaling of the system with additional sensors; (v) allows robust sensor fusion algorithms implementation.

Summarising the direction chosen in this dissertation is.

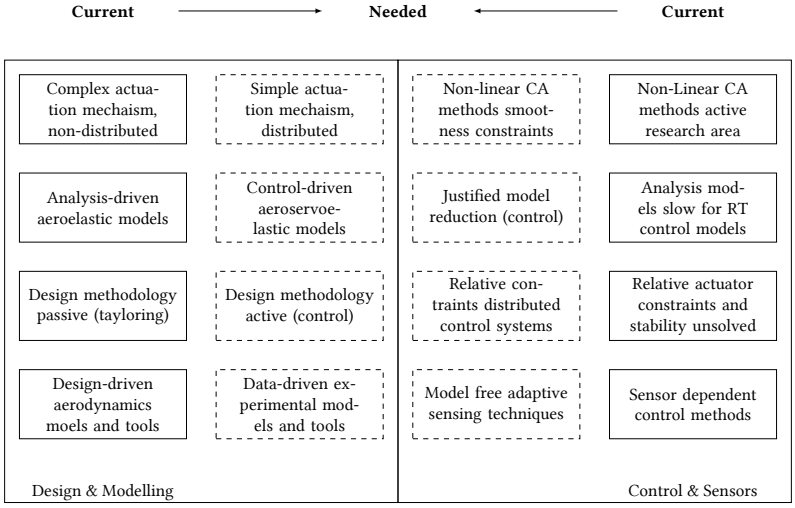


Figure 2.4: Overview of identified gaps in the literature relevant to smart morphing wings.

2.6. Integration of the Concepts

A review of the current state-of-the-art relevant to the smart morphing wing has helped identify gaps in design, modelling, sensing, control and integration domains. The relevant gaps are summarised in Fig. 2.4. Based on these findings, the rationale behind the smart morphing wing, developed during this dissertation, is conceptualised. The concept of integrating the discussed components is presented in Fig. 2.5. The smart morphing wing is displayed with the continuously morphing trailing edge in the centre. Several key components of the diagram are:

- Integrated actuators allow morphing of trailing edge surface defined by $S_\delta(y)$.
- Sensors are installed in the wing to allow real-time measurements of (i) shape of the wing with fibre-optic sensors, (ii) boundary layer sensing with piezoelectric sensors for in-flight cruise shape optimisation and automated high-lift generation (iii) high-speed cameras installed at the root to provide shape feedback of the morphing surfaces to the controller.
- A multi-objective optimal controller adaptively drives the actuator inputs, $S_{c_\delta}(y)$, towards the desired shape $S_\delta(y)$ curve in order to optimise in real-time for several objectives formulated in the objective function J.
- As the wing encounters gust and varying atmospheric conditions, the controller continuously adapts the morphing surfaces to obtain the optimal lift distribution for the given objectives, $f_A(t) - f_B(t)$.

Knowledge of the state of the boundary layer is important for shape control both for in-flight cruise shape optimisation and automated high-lift generation. Knowing whether the boundary layer is turbulent or laminar is important for cruise shape optimisation, while knowing whether the boundary layer is attached or separated is important in the case of automated high-lift. The flow sensing hardware must be integrated into the wing skin since it must be able to operate in flight.

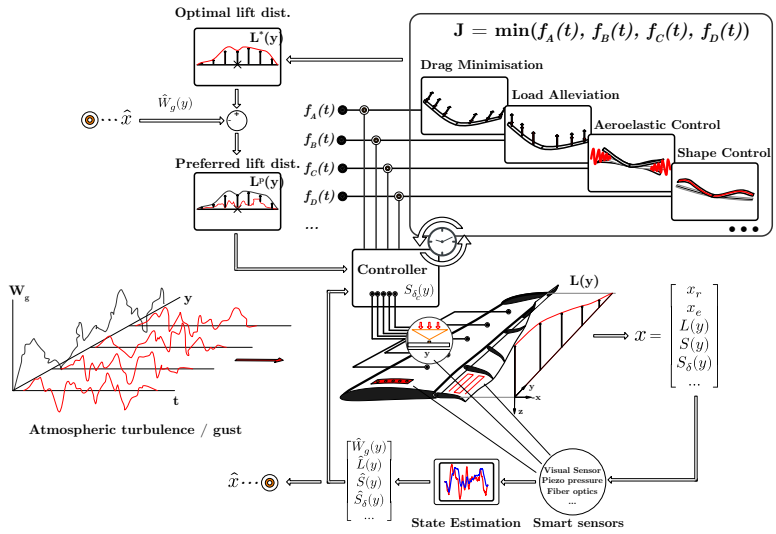


Figure 2.5: Concept integration of the smart morphing wing with synchronised multi-objective control allocation and smart sensing.

2.7. ROADMAP

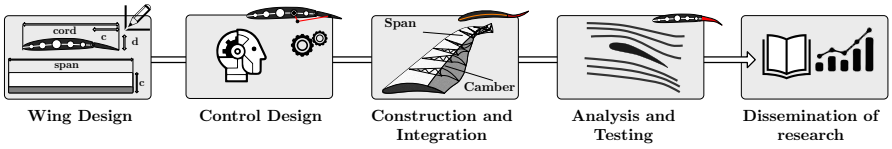


Figure 2.6: Timeline of the dissertation.

The timeline of this dissertation, presented in Fig. 2.6, is focused on the end goal, which is the development and the assessment of the smart morphing wing demonstrator wind tunnel studies. The goal is envisioned to be reached through several incremental studies in morphing wing design, control, sensing, integration and testing.

The roadmap of this dissertation, presented in Fig. 2.7, uses the runway analogy. Here, the development is commenced through (A) preliminary sizing and design studies, followed by (B) numerical evaluation of the designs. The benefits of the smart morphing

wing are investigated with incremental design improvements and assessment of (C) discretised, distributed control surface design and (D) trailing edge morphing. The two design approaches are assessed in terms of their ability to meet real-time objectives. Final recommendations are made on the feasibility of (E) full morphing smart wing concept and suggestions for adopting this technology in future sustainable aircraft designs.

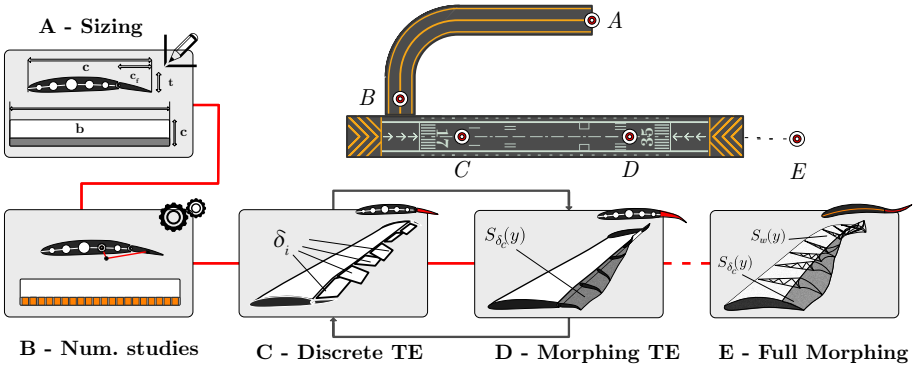


Figure 2.7: Roadmap of the dissertation.

REFERENCES

[1] D. Lentink, U. K. Müller, E. J. Stamhuis, R. de Kat, W. van Gestel, L. L. M. Veldhuis, P. Henningsson, A. Hedenström, J. J. Videler, and J. L. van Leeuwen, "How swifts control their glide performance with morphing wings," *Nature*, vol. 446, pp. 1082–1085, Apr. 2007.

[2] P. Henningsson, A. Hedenström, and R. J. Bomphrey, "Efficiency of Lift Production in Flapping and Gliding Flight of Swifts," *PLoS ONE*, vol. 9, p. e90170, Feb. 2014.

[3] H. R. Jex and F. E. C. Culick, "Flight Control Dynamics of the 1903 Wright Flyer," in *12th Atmospheric Flight Mechanics Conference*, (Reston, Virigina), pp. 534–548, American Institute of Aeronautics and Astronautics, Aug. 1985.

[4] T. A. Weisshaar, "Morphing Aircraft Systems: Historical Perspectives and Future Challenges," *Journal of Aircraft*, vol. 50, pp. 337–353, Mar. 2013.

[5] S. Barbarino, O. Bilgen, R. M. Ajaj, M. I. Friswell, and D. J. Inman, "A Review of Morphing Aircraft," *Journal of Intelligent Material Systems and Structures*, vol. 22, no. 9, pp. 823–877, 2011.

[6] M. Kintscher, S. Geier, H. P. Monner, and M. Wiedemann, "Investigation of multi-material laminates for smart droop nose devices," *29th Congress of the International Council of the Aeronautical Sciences, ICAS*, pp. 1–11, 2014.

[7] J. Sodja, M. J. Martinez, J. C. Simpson, and R. De Breuker, "Experimental evaluation of a morphing leading edge concept," *Journal of Intelligent Material Systems and Structures*, vol. 30, pp. 2953–2969, Nov. 2019.

[8] S. Vasista, A. De Gaspari, S. Ricci, J. Riemenschneider, H. P. Monner, and B. Van De Kamp, "Compliant structures-based wing and wingtip morphing devices," *Aircraft Engineering and Aerospace Technology*, vol. 88, pp. 311–330, Mar. 2016.

[9] F. Previtali, A. F. A. Arrieta, and P. Ermanni, "Performance of a Three-Dimensional Morphing Wing and Comparison with a Conventional Wing," *AIAA Journal*, vol. 52, pp. 2101–2113, Oct. 2014.

[10] G. Molinari, M. Quack, A. F. Arrieta, M. Morari, and P. Ermanni, "Design, realization and structural testing of a compliant adaptable wing," *Smart Materials and Structures*, vol. 24, p. 105027, Oct. 2015.

[11] F. Previtali and P. Ermanni, "Performance of a non-tapered 3D morphing wing with integrated compliant ribs," *Smart Materials and Structures*, vol. 21, no. 5, 2012.

[12] G. Molinari, A. F. Arrieta, and P. Ermanni, "Aero-Structural Optimization of Three-Dimensional Adaptive Wings with Embedded Smart Actuators," *AIAA Journal*, vol. 52, pp. 1940–1951, Sept. 2014.

[13] B. K. S. Woods, I. Dayyani, and M. I. Friswell, "Fluid/Structure-Interaction Analysis of the Fish-Bone-Active-Camber Morphing Concept," *Journal of Aircraft*, vol. 52, pp. 307–319, Jan. 2015.

[14] B. K. Woods, O. Bilgen, and M. I. Friswell, "Wind tunnel testing of the fish bone active camber morphing concept," *Journal of Intelligent Material Systems and Structures*, vol. 25, pp. 772–785, May 2014.

[15] B. K. S. Woods, M. I. Friswell, and N. M. Wereley, "Advanced Kinematic Tailoring for Morphing Aircraft Actuation," *AIAA Journal*, vol. 52, pp. 788–798, Apr. 2014.

[16] S. Kota, J. A. Hetrick, R. Osborn, D. Paul, E. Pendleton, P. Flick, and C. Tilmann, "Design and application of compliant mechanisms for morphing aircraft structures," (San Diego, CA), p. 24, SPIE, Aug. 2003.

[17] S. Kota, P. Flick, and F. Collier, "Flight testing of the FlexFloil™ adaptive compliant trailing edge," *54th AIAA Aerospace Sciences Meeting*, 2016.

[18] B. Jenett, S. Calisch, D. Cellucci, N. Cramer, N. Gershenfeld, S. Swei, and K. C. Cheung, "Digital Morphing Wing: Active Wing Shaping Concept Using Composite Lattice-Based Cellular Structures," *Soft Robotics*, vol. 4, no. 1, pp. 33–48, 2017.

[19] N. B. Cramer, D. W. Cellucci, O. B. Formoso, C. E. Gregg, B. E. Jenett, J. H. Kim, M. Lendraitis, S. S. Swei, G. T. Trinh, K. V. Trinh, and K. C. Cheung, "Elastic shape morphing of ultralight structures by programmable assembly," *Smart Materials and Structures*, vol. 28, p. 055006, May 2019.

[20] D. Keidel, U. Fasel, and P. Ermanni, "Concept Investigation of a Lightweight Composite Lattice Morphing Wing," *AIAA Journal*, vol. 59, pp. 2242–2250, Jan. 2021. Publisher: American Institute of Aeronautics and Astronautics.

[21] N. Nguyen, S. Lebofsky, E. Ting, U. Kaul, D. Chaparro, and J. Urnes, "Development of Variable Camber Continuous Trailing Edge Flap for Performance Adaptive Aeroelastic Wing," pp. 2015–01–2565, Sept. 2015.

[22] Y. Ferrier, N. T. Nguyen, E. Ting, D. Chaparro, X. Wang, C. C. de Visser, and Q. P. Chu, "Active Gust Load Alleviation of High-Aspect Ratio Flexible Wing Aircraft," in *2018 AIAA Guidance, Navigation, and Control Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, Jan. 2018.

[23] N. T. Nguyen, E. Ting, D. Chaparro, M. C. Drew, and S. S.-M. Swei, "Multi-Objective Flight Control for Drag Minimization and Load Alleviation of High-Aspect Ratio Flexible Wing Aircraft," in *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, (Grapevine, Texas), American Institute of Aeronautics and Astronautics, Jan. 2017.

[24] N. Werter, J. Sodja, G. Spirlet, and R. De Breuker, "Design and Experiments of a Warp Induced Camber and Twist Morphing Leading and Trailing Edge Device," in *24th AIAA/AHS Adaptive Structures Conference*, (San Diego, California, USA), American Institute of Aeronautics and Astronautics, Jan. 2016.

[25] X. Wang, T. Mkhoyan, and R. De Breuker, "Nonlinear incremental control for flexible aircraft trajectory tracking and load alleviation," *AIAA Scitech 2021 Forum*, pp. 1–19, aug 2021.

[26] M. d. F. V. Pereira, I. Kolmanovsky, C. E. S. Cesnik, and F. Vetrano, "Model Predictive Control Architectures for Maneuver Load Alleviation in Very Flexible Aircraft," in *AIAA Scitech 2019 Forum*, (San Diego, California), American Institute of Aeronautics and Astronautics, Jan. 2019.

[27] S. Haghighat, H. H. T. Liu, and J. R. R. A. Martins, "Model-Predictive Gust Load Alleviation Controller for a Highly Flexible Aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 35, pp. 1751–1766, Nov. 2012.

[28] E. Vartio, E. Shaw, and T. Vetter, "Gust Load Alleviation Flight Control System Design for a SensorCraft Vehicle," in *26th AIAA Applied Aerodynamics Conference*, (Honolulu, Hawaii), pp. 1–10, American Institute of Aeronautics and Astronautics, Aug. 2008. Issue: August.

[29] J.-W. van Wingerden, A. Hulskamp, T. Barlas, I. Houtzager, H. Bersee, G. van Kuik, and M. Verhaegen, "Two-Degree-of-Freedom Active Vibration Control of a Prototyped "Smart" Rotor," *IEEE Transactions on Control Systems Technology*, vol. 19, pp. 284–296, Mar. 2011. Publisher: IEEE.

[30] Y. Bi, C. Xie, C. An, and C. Yang, "Gust load alleviation wind tunnel tests of a large-aspect-ratio flexible wing with piezoelectric control," *Chinese Journal of Aeronautics*, vol. 30, pp. 292–309, Feb. 2017.

[31] D. H. Baldelli, D.-H. Lee, R. S. S. Pena, and B. Cannon, "Modeling and Control of an Aeroelastic Morphing Vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 31, pp. 1687–1699, Nov. 2008.

[32] N. T. Nguyen and E. A. Tal, "A Multi-Objective Flight Control Approach for Performance Adaptive Aeroelastic Wing," Jan. 2015.

[33] N. T. Nguyen, K. E. Hashemi, and M. C. Drew, "Multi-Objective Adaptive Control for Load Alleviation and Drag Minimization of Flexible Aircraft," in *2018 AIAA Guidance, Navigation, and Control Conference*, (Kissimmee, Florida), American Institute of Aeronautics and Astronautics, Jan. 2018.

[34] X. Wang, E.-J. Van Kampen, Q. P. Chu, and P. Lu, "Stability Analysis for Incremental Nonlinear Dynamic Inversion Control," in *2018 AIAA Guidance, Navigation, and Control Conference*, (Kissimmee, Florida), American Institute of Aeronautics and Astronautics, Jan. 2018.

[35] Q. Shen, D. Wang, S. Zhu, and K. Poh, "Finite-time fault-tolerant attitude stabilization for spacecraft with actuator saturation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, pp. 2390–2405, July 2015.

[36] N. Vafamand, M. H. Asemani, A. Khayatiyan, M. H. Khooban, and T. Dragičević, "TS fuzzy model-based controller design for a class of nonlinear systems including nonsmooth functions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 233–244, 2020. Publisher: IEEE.

[37] Z. Lin, S. Lin, S. Wu, G. Ma, and Z. Liang, "Vibration control of a flexible spacecraft system with input backlash," *IEEE Access*, vol. 7, pp. 87017–87026, 2019. Publisher: IEEE.

[38] L. Sanches, T. A. Guimarães, and F. D. Marques, "Aeroelastic tailoring of nonlinear typical section using the method of multiple scales to predict post-flutter stable LCOs," *Aerospace Science and Technology*, vol. 90, pp. 157–168, July 2019.

[39] X. Wang, E. Van Kampen, Q. P. Chu, and R. De Breuker, "Flexible Aircraft Gust Load Alleviation with Incremental Nonlinear Dynamic Inversion," *Journal of Guidance, Control, and Dynamics*, vol. 42, pp. 1519–1536, July 2019.

[40] R. Tekin, K. S. Erer, and F. Holzapfel, "Control of Impact Time with Increased Robustness via Feedback Linearization," *Journal of Guidance, Control, and Dynamics*, vol. 39, pp. 1682–1689, July 2016.

[41] A. N. Kalliny, A. A. El-Badawy, and S. M. Elkhamisy, "Command-Filtered Integral Backstepping Control of Longitudinal Flapping-Wing Flight," *Journal of Guidance, Control, and Dynamics*, vol. 41, pp. 1556–1568, July 2018.

[42] S. Sieberling, Q. P. Chu, and J. A. Mulder, "Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction,"

[43] P. Acquatella, E.-J. Van Kampen, and Q. P. Chu, "Incremental Backstepping for Robust Nonlinear Flight Control," in *EuroGNC 2013, 2nd CEAS Specialist Conference on Guidance, Navigation & Control*, (Delft University of Technology, Delft, The Netherlands), 2013.

[44] F. Grondman, G. Looye, R. O. Kuchar, Q. P. Chu, and E. van Kampen, "Design and Flight Testing of Incremental Nonlinear Dynamic Inversion-based Control Laws for a Passenger Aircraft," in *2018 AIAA Guidance, Navigation, and Control Conference*, (Kissimmee, Florida), American Institute of Aeronautics and Astronautics, Jan. 2018. Issue: January.

[45] S. Sun, X. Wang, Q. P. Chu, and C. C. de Visser, "Incremental Nonlinear Fault-Tolerant Control of a Quadrotor With Complete Loss of Two Opposing Rotors," *IEEE Transactions on Robotics*, pp. 1–15, 2020.

[46] X. Wang, E.-J. v. Kampen, Q. Chu, and P. Lu, "Incremental Sliding-Mode Fault-Tolerant Flight Control," *Journal of Guidance, Control, and Dynamics*, vol. 42, pp. 244–259, Feb. 2019.

[47] I. Matamoros and C. C. de Visser, "Incremental Nonlinear Control Allocation for a Tailless Aircraft with Innovative Control Effectors," in *2018 AIAA Guidance, Navigation, and Control Conference*, (Kissimmee, Florida), pp. 1–25, American Institute of Aeronautics and Astronautics, Jan. 2018.

[48] C. C. de Visser, E. Brunner, and M. Verhaegen, "On distributed wavefront reconstruction for large-scale adaptive optics systems," *Journal of the Optical Society of America A*, vol. 33, p. 817, May 2016.

[49] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, pp. 2699–2704, Oct. 2012.

[50] T. Dierks and S. Jagannathan, "Online Optimal Control of Affine Nonlinear Discrete-Time Systems With Unknown Internal Dynamics by Using Time-Based Policy Update," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1118–1129, July 2012.

[51] C. Peng and J. Ma, "Online integral reinforcement learning control for an uncertain highly flexible aircraft using state and output feedback," *Aerospace Science and Technology*, vol. 109, p. 106442, Feb. 2021. Publisher: Elsevier Masson.

[52] M. Kumar, K. Rajagopal, S. N. Balakrishnan, and N. T. Nguyen, "Reinforcement learning based controller synthesis for flexible aircraft wings," *Automatica Sinica, IEEE/CAA Journal of*, vol. 1, no. 4, pp. 435–448, 2014.

[53] J. Ma and C. Peng, "Adaptive model-free fault-tolerant control based on integral reinforcement learning for a highly flexible aircraft with actuator faults," *Aerospace Science and Technology*, vol. 119, p. 107204, Dec. 2021.

[54] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, pp. 477–484, Feb. 2009. Publisher: Pergamon.

[55] L. Obringer, "Control of Flexible Structures," 2017.

[56] C. E. S. Cesnik, P. J. Senatore, W. Su, E. M. Atkins, and C. M. Shearer, "X-HALE: A very flexible unmanned aerial vehicle for nonlinear aeroelastic tests," *AIAA Journal*, vol. 50, no. 12, pp. 2820–2833, 2012.

[57] A. W. Burner and T. Liu, "Videogrammetric model deformation measurement technique," *Journal of Aircraft*, vol. 38, no. 4, pp. 745–754, 2001.

[58] P. I. Corke, "Visual Control of Robot Manipulators – a Review," in *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, pp. 1–31, World Scientific, 1993.

[59] Y. Choi, M. Martel, S. I. Briceno, and D. N. Mavris, "Multi-UAV Trajectory Optimization and Deep Learning-based Imagery Analysis for a UAS-based Inventory Tracking Solution," in *AIAA Scitech 2019 Forum*, (San Diego, California), American Institute of Aeronautics and Astronautics, Jan. 2019.

[60] Y. Hu, Y. Cao, M. Ding, and L. Zhuang, "Airport Detection for Fixed-Wing Unmanned Aerial Vehicle Landing Using a Hierarchical Architecture," *Journal of Aerospace Information Systems*, vol. 16, pp. 214–223, June 2019.

[61] K. Abu-Jbara, G. Sundaramorthi, and C. Claudel, "Fusing Vision and Inertial Sensors for Robust Runway Detection and Tracking," *Journal of Guidance, Control, and Dynamics*, vol. 41, pp. 1929–1946, Sept. 2018.

[62] P. Agrawal, A. Ratnoo, and D. Ghose, "Image Segmentation-Based Unmanned Aerial Vehicle Safe Navigation," *Journal of Aerospace Information Systems*, vol. 14, pp. 391–410, July 2017.

[63] J. Valasek, D. Famularo, and M. Marwaha, "Fault-Tolerant Adaptive Model Inversion Control for Vision-Based Autonomous Air Refueling," *Journal of Guidance, Control, and Dynamics*, vol. 40, pp. 1336–1347, June 2017.

[64] C. Parsons, Z. Paulson, S. Nykl, W. Dallman, B. G. Woolley, and J. Pecarina, "Analysis of Simulated Imagery for Real-Time Vision-Based Automated Aerial Refueling," *Journal of Aerospace Information Systems*, vol. 16, pp. 77–93, Mar. 2019. Publisher: American Institute of Aeronautics and Astronautics Inc.

[65] R. D. Abousleiman, O. A. Rawashdeh, and M. R. Siadat, "Statistical algorithm for attitude estimation from real-time aerial video," *Journal of Aerospace Computing, Information and Communication*, vol. 7, no. 10, pp. 322–337, 2010.

[66] H. P. J. Veerman, H. Kannemans, and H. W. Jentink, "High Accuracy In-Flight Wing Deformation Measurements based on Optical Correlation Technique," Technical Report NLR-TP-2008-718, National Aerospace Center NLR, 2009.

[67] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive Color Attributes for Real-Time Visual Tracking," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1090–1097, IEEE, June 2014.

[68] P. Tchoryk Jr, C. B. Watkins, S. K. Lindemann, P. B. Hays, and C. A. Nardell, "Molecular optical air data system (MOADS)," in *Laser Radar Technology and Applications VI*, vol. 4377, pp. 194–205, SPIE, 2001.

[69] M. J. Verbeek and H. W. Jentink, "Optical Air Data System Flight Testing," Technical Report NLR-TP-2012-068, National Aerospace Center NLR, 2012.

[70] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern Recognition Letters*, vol. 34, pp. 3–19, Jan. 2013.

[71] A. N. Belbachir, *Smart cameras*, vol. 2. Springer, 2010. ISSN: 1666-6038 Publication Title: Smart Cameras.

[72] NVIDIA, "High Performance AI at the Edge \textbar NVIDIA Jetson TX2."

[73] M. H. Shirk, T. J. Hertz, and T. A. Weisshaar, "Aeroelastic tailoring - Theory, practice, and promise," *Journal of Aircraft*, vol. 23, pp. 6–18, Jan. 1986.

[74] T. Weisshaar, "Aeroelastic tailoring - Creative uses of unusual materials," in *28th Structures, Structural Dynamics and Materials Conference*, (Monterey,CA,U.S.A.), American Institute of Aeronautics and Astronautics, Apr. 1987.

[75] R. De Breuker, *Energy-based Aeroelastic Analysis and Optimisation of Morphing Wings*. PhD Thesis, Delft University of Technology, 2011.

[76] N. Werter, *Aeroelastic Modelling and Design of Aeroelastically Tailored and Morphing Wings*. PhD thesis, Delft University of Technology, 2017.

[77] M. Natella, *Aeroelastic Tailoring of Composite Aircraft*. PhD Thesis, Delft University of Technology, 2020.

[78] J. Katz and A. Plotkin, *Low-Speed Aerodynamics*. Cambridge University Press, 2 ed., Feb. 2001.

[79] R. Palacios, J. Murua, and R. Cook, "Structural and Aerodynamic Models in Nonlinear Flight Dynamics of Very Flexible Aircraft," *AIAA Journal*, vol. 48, no. 11, pp. 2648–2659, 2010.

[80] M. Ritter, P. Teixeira, and C. E. S. Cesnik, "Comparison of Nonlinear Aeroelastic Methods for Maneuver Simulation of Very Flexible Aircraft," in *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, (Kissimmee, Florida), American Institute of Aeronautics and Astronautics, Jan. 2018.

[81] M. J. Patil and D. H. Hodges, "On the importance of aerodynamic and structural geometrical nonlinearities in aeroelastic behavior of high-aspect-ratio wings," *Journal of Fluids and Structures*, vol. 19, pp. 905–915, Aug. 2004.

[82] B. J. Brelje and J. R. Martins, "Electric, hybrid, and turboelectric fixed-wing aircraft: A review of concepts, models, and design approaches," *Progress in Aerospace Sciences*, vol. 104, pp. 1–19, Jan. 2019.

[83] J. G. Leishman, "Subsonic unsteady aerodynamics caused by gusts using the indicial method," *Journal of Aircraft*, vol. 33, pp. 869–879, Sept. 1996.

[84] R. De Breuker, S. Binder, and A. Wildschek, "Combined Active and Passive Loads Alleviation through Aeroelastic Tailoring and Control Surface/Control System Optimization," in *2018 AIAA Aerospace Sciences Meeting*, (Kissimmee, Florida), American Institute of Aeronautics and Astronautics, Jan. 2018.

[85] F. J. Silvestre and R. Luckner, "Experimental Validation of a Flight Simulation Model for Slightly Flexible Aircraft," *AIAA Journal*, vol. 53, pp. 3620–3636, Dec. 2015.

[86] E. Albano and W. P. Rodden, "A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flows.," *AIAA Journal*, vol. 7, pp. 279–285, Feb. 1969.

[87] J. Murua, R. Palacios, and J. M. R. Graham, "Applications of the unsteady vortex-lattice method in aircraft aeroelasticity and flight dynamics," *Progress in Aerospace Sciences*, vol. 55, pp. 46–72, Nov. 2012.

[88] P. Geuzaine, G. Brown, C. Harris, and C. Farhat, "Aeroelastic Dynamic Analysis of a Full F-16 Configuration for Various Flight Conditions," *AIAA Journal*, vol. 41, pp. 363–371, Mar. 2003.

[89] L. Reimer, M. Ritter, R. Heinrich, and W. Krüger, "CFD-based Gust Load Analysis for a Free-flying Flexible Passenger Aircraft in Comparison to a DLM-based Approach," in *22nd AIAA Computational Fluid Dynamics Conference*, p. 2455, American Institute of Aeronautics and Astronautics, 2015.

[90] N. P. M. Werter, R. De Breuker, and M. M. Abdalla, "Continuous-Time State-Space Unsteady Aerodynamic Modeling for Efficient Aeroelastic Loads Analysis," in *IFASD 2015: 16th International Forum on Aeroelasticity and Structural Dynamics, Saint Petersburg, Russia, 28 June-2 July 2015*, 2015.

[91] K. C. Hall, "Eigenanalysis of unsteady flows about airfoils, cascades, and wings," *AIAA Journal*, vol. 32, pp. 2426–2432, Dec. 1994.

[92] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE transactions on automatic control*, vol. 26, no. 1, pp. 17–32, 1981. Publisher: IEEE.

[93] C. W. Rowley, "Model Reduction for Fluids, Using Balanced Proper Orthogonal Decomposition," *International Journal of Bifurcation and Chaos*, vol. 15, no. 03, pp. 997–1013, 2005. Publisher: World Scientific.

[94] K. Willcox and J. Peraire, "Balanced Model Reduction via the Proper Orthogonal Decomposition," *AIAA Journal*, vol. 40, pp. 2323–2330, Nov. 2002.

[95] E. Gillebaart and R. De Breuker, "Reduced-order modeling of continuous-time state-space unsteady aerodynamics," *53rd AIAA Aerospace Sciences Meeting*, 2015.

[96] H. J. Tol, M. Kotsonis, C. C. de Visser, and B. Bamieh, "Localised estimation and control of linear instabilities in two-dimensional wall-bounded shear flows," *Journal of Fluid Mechanics*, vol. 824, pp. 818–865, Aug. 2017.

[97] B. Besselink, N. van de Wouw, J. M. A. Scherpen, and H. Nijmeijer, "Model Reduction for Nonlinear Systems by Incremental Balanced Truncation," *IEEE Transactions on Automatic Control*, vol. 59, pp. 2739–2753, Oct. 2014.

[98] S. Gugercin and A. C. Antoulas, "A Survey of Model Reduction by Balanced Truncation and Some New Results," *International Journal of Control*, vol. 77, pp. 748–766, May 2004.

[99] C. Poussot-Vassal, D. Quero, and P. Vuillemin, "Data-driven approximation of a high fidelity gust-oriented flexible aircraft dynamical model," *IFAC-PapersOnLine*, vol. 51, pp. 559–564, Jan. 2018. Publisher: Elsevier.

[100] C. Alba, A. Elham, B. J. German, and L. L. L. M. Veldhuis, "A surrogate-based multi-disciplinary design optimization framework modeling wing–propeller interaction," *Aerospace Science and Technology*, vol. 78, pp. 721–733, July 2018. Publisher: Elsevier Masson.

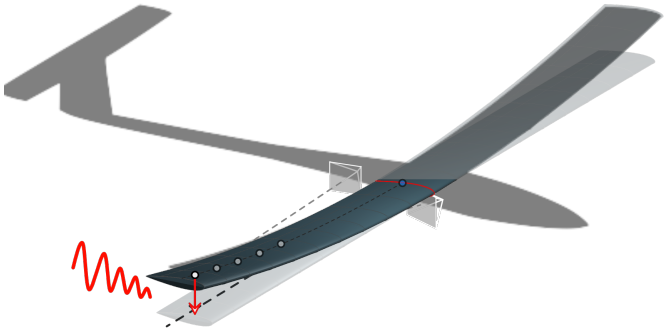
[101] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic Mode Decomposition with Control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, pp. 142–161, Jan. 2016.

[102] N. Fonzi, S. L. Brunton, and U. Fasel, "Data-driven nonlinear aeroelastic models of morphing wings for control," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 476, p. 20200079, July 2020.

[103] E. Morelli, "In-flight system identification," in *23rd Atmospheric Flight Mechanics Conference*, (Boston,MA,U.S.A.), American Institute of Aeronautics and Astronautics, Aug. 1998.

I

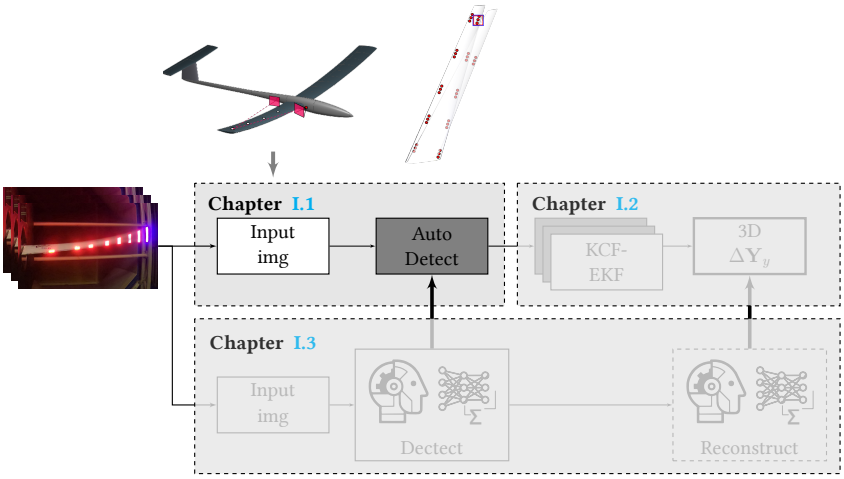
Smart Sensing: Visual Tracking



I.1

Adaptive Real-Time Clustering for Visual Tracking

In this Chapter, an image tracking pipeline is developed using a robust machine learning approach, aiming to (i) automatically label visual markers and (ii) investigate a non-invasive state estimation approach for online control applications of flexible aircraft wings. This Chapter, along with Chapters I.2 and I.3, is part of the first wind-tunnel campaign aiming to investigate a smart vision-based sensing approach for morphing and flexible wings [1].



This Chapter is based on the following journal paper:
T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive Real-Time Clustering Method for Dynamic Visual Tracking of Very Flexible Wings," *Journal of Aerospace Information Systems*, vol. 18, pp. 58–79, jan 2021

I.1

Advancements in aircraft controller design, paired with increasingly flexible aircraft concepts, create the need to develop novel (smart) adaptive sensing methods suitable for aeroelastic state estimation. A potentially universal and non-invasive approach is visual tracking. However, many tracking methods require manual selection of initial marker locations at the start of a tracking sequence. This study addresses the gap by investigating a robust machine learning approach for unsupervised automatic labelling of visual markers. The method utilises fast Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and adaptive image segmentation pipeline with HSV colour filter to extract and label the marker centres under the presence of marker failure. A comparative study assesses the DBSCAN clustering performance against an alternative clustering method, the Disjoint-set data structure. The segmentation-clustering pipeline with DBSCAN is capable of running in real-time at 250 fps on a single camera image sequence with a resolution of 1088×600 pixels. To increase robustness against noise, a novel formulation — the inverse DBSCAN, DBSCAN^{-1} — is introduced. This approach is validated on an experimental dataset collected from camera observations of a flexible wing undergoing gust excitations in a wind tunnel, demonstrating an excellent match with the ground truth obtained with a laser vibrometer measurement system.

Nomenclature

A, B	=	Subsets of dataset D
$P(x,\ y)$	=	Density distribution of particles (2D)
$B(x', y')$	=	Kernel matrix
$\mathbf{P}(x, y)$	=	Cloud of cluster centres
$\bar{\mathbf{c}}_p$	=	Centroid of points $\mathbf{P}$
$\mathbf{P}_{\theta hull}$	=	Convex radial hull
$\bar{\mathbf{c}}_{cp}$	=	Centroid of cluster centres
$p,\ q$	=	Scatter points
D	=	Dataset
$p_n,\ q_n$	=	Scatter noise particles
$dist\,(p, q)_{euclid}$	=	Euclidean distance function
$R_{yy}(\tau)$	=	Auto Correlation Function
$f(I(x, y))$	=	Filtering (sequence) operation
$S_{yy}(\omega)$	=	Auto Power Spectral Density
$f_{dilate}(I(x, y))$	=	Dilate operation
V_∞	=	Wind tunnel flow velocity
$f_{erode}(I(x, y))$	=	Erode operation
$w_1,\ w_2$	=	Class variance weights (Otsu)
f_g	=	Gust vane frequency
$Z_\epsilon(p_n)$	=	The ϵ neighbourhood of noise points p_n
$f_{morph}(I(x, y))$	=	Morphological operations (combined)
z	=	Greyscale value
f_{norm}	=	Global normalisation operation
α_g	=	Gust vane angle
$G_f(x, y)$	=	Filtered image
γ	=	Radius tolerance
$I(x, y)$	=	Input image
ϵ	=	Radius of neighbouring points
$I(z)$	=	Gaussian Noise probability density
$\boldsymbol{\theta}_{cp}$	=	Vector angles from centroid to marker
$J(x, y)$	=	Noise input image

μ	=	Mean of $I(z)$ distribution
$MaxPts$	=	DBSCAN^{-1} max points dense region
μ_{dst}	=	Mean of the cluster population
$MinPts$	=	DBSCAN min points dense region
μ_I	=	Mean of points in 2D image
m_i	=	Cluster centre size
σ	=	Standard deviation of $I(z)$ distribution
$N_\epsilon(p)$	=	The ϵ neighbourhood of points p
σ_{dst}	=	Standard deviation of cluster population
$N(x,\, y,\, t)$	=	Random seed initialised noise mask
σ_I	=	Standard deviation points in 2D image
n_i	=	Cluster population size
$\sigma_w^2(\tau_{th})$	=	Intra-class variance (Otsu)
$n_{i_{noise}}$	=	Noise particle population size
$\sigma_1^2,\, \sigma_2^2$	=	Class variances (Otsu)
$O(...)$	=	Computational complexity
τ_{th}	=	Threshold parameter

I.1.1. Introduction

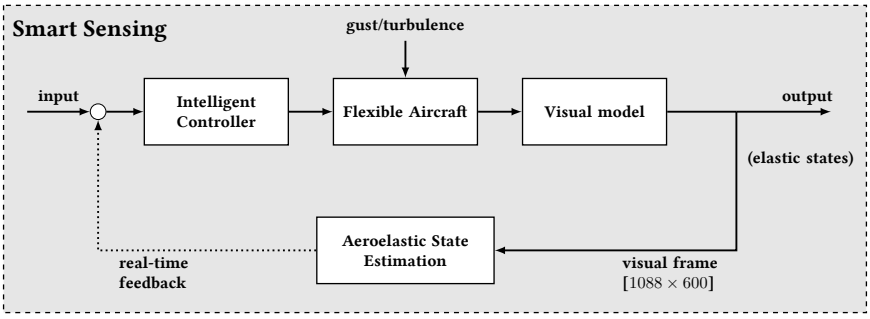


Figure I.1.1: State estimation setup using visual tracking.

In the context of aeroservoelastic control, monitoring the entire wingspan can be crucial for proper delegation of control actions. This objective may involve installing many conventional accelerometers that are likely subject to noise and bias, must deal with certification requirements, or might face challenges associated with correct geometric placement or limited mounting space. A smart sensing approach is desired for those examples of wing structures that rely on novel types of sensors for providing feedback to an intelligent controller.

A solution that can significantly reduce the complexity associated with hardware installation and provide the flexibility needed for employing novel state estimation techniques is aeroelastic state estimation by visual methods. A schematic of aeroelastic state estimation using vision — consisting of an intelligent controller, the aircraft model and the visual model — is illustrated in Fig. I.1.1. This Chapter aims to contribute to the aeroelastic state estimation block such that the control loop can be closed with the dotted line.

The use of visual information for observing deformations has been successfully im-

plemented on wind tunnel models in early studies [2], and has also seen widespread application in robot manipulation [3]. However, in recent years, the capability in terms of onboard computation and camera quality has immensely increased, while the hardware has become more compact [4, 5]. These developments open the door for numerous embedded applications using a camera as a sensor for aircraft systems.

Various vision applications for areal systems have been investigated for areal imagery in recent studies [6–8]. Vision-based information can also be used for aerial navigation [9] and flight control tasks, such as aerial refuelling [10], landing [11] and estimation of rigid body aircraft states, such as altitude [12]. However, within the scope of flexible and morphing aircraft systems —as these systems are more prone to exhibiting higher responses to aerodynamic loads—, the challenge lies in estimating the impact of the flexibility on the dynamics of the system, which cannot always be accounted for in the early design stage. The study by Weisshaar et al. [13], highlights the ability to monitor and communicate structural state information as one of the key aspects of the development of smart morphing structures. Vision-based feedback systems can play a crucial role in this task as one camera system can observe multiple nodes of the system's flexible states in a sequence of images [14]. In particular, fuselage-mounted camera systems can provide significant advantages for flexible aircraft systems, save costs associated with installation and certification, and have the potential of being non-invasive and universally applicable. Vision-based information has been shown to be suitable for direct real-time feedback of flexible states of an aircraft [15, 16].

Sequenced image data is also a rich source of information: data collected over an extended period unlocks the opportunity to approach the state estimation from a new perspective using machine learning methods. One of the key challenges is the need for robust, unsupervised and computationally efficient clustering methods. Several studies investigated the performance of clustering methods by using improved [17, 18] and parallel DBSCAN methods [19]. However, a gap remains for a streamlined approach to unsupervised clustering with robustness against noise. In particular, while many suitable tracking methods exist for marker detection, correctly labelling the initial markers in the visual frame is still not a trivial task [20].

In this Chapter, two machine learning methods were implemented for unsupervised clustering of marker labels, meaning that they do not require the number of clusters and initial guesses as input. The sequence of images is filtered with two image segmentation approaches to obtain a mask for clustering operations. A comparison was made between the two machine learning methods, DBSCAN [21] and Disjoint-set data structure [22], and a segmentation-clustering pipeline was developed based on Hue-Saturation-Value (HSV) [23] and adaptive thresholding with Otsu's method [24].

A novel approach to DBSCAN — the inverse DBSCAN (DBSCAN^{-1}) — was introduced and implemented in the study. In this approach, the clustering problem is reformulated into a noise filtering problem, and an additional parameter, $MaxPts$, is introduced into the formulation. The crux of DBSCAN^{-1} lies in isolating the group of desired clusters and classifying them as *noise*, i.e. points surrounded by *too many other points* (filtered by max $MaxPts$ condition). Subsequently, the desired clusters of points are rejected as noise, while the *true noise* is identified explicitly and removed from the dataset in a follow-up step.

For the purpose of investigating the robustness of the method, the input images were subjected to Gaussian noise, and both the nominal DBSCAN as well as DBSCAN^{-1} were assessed in performance with less noise filtering. An image tracking pipeline was developed to test this clustering method on an image sequence. It was observed that the proposed method is capable of real-time tracking and achieving speeds of 250+ fps (frames-per-second), measured on an image sequence of a single camera with a resolution of 1088×600 pixels in a laboratory environment on a standard Dell Optiplex 7400 and a 2.3 GHz Intel Core i5 16G MacBook. Hence, the method is suitable for online control applications.

The approach was tested on an image sequence of a flexible wing equipped with LED (light-emitting diode) markers, undergoing oscillatory motion under gust excitation in the Open Jet Facility (OJF) wind tunnel of the Delft University of Technology. Furthermore, the effect of the frequency content was studied to investigate a potential implementation in the pipeline for adjusting the segmentation and clustering parameters. A schematic of the experimental setup is shown in Fig. I.1.2; in this experiment, the same gust generator was used as the one developed for OJF in a previous study [25].

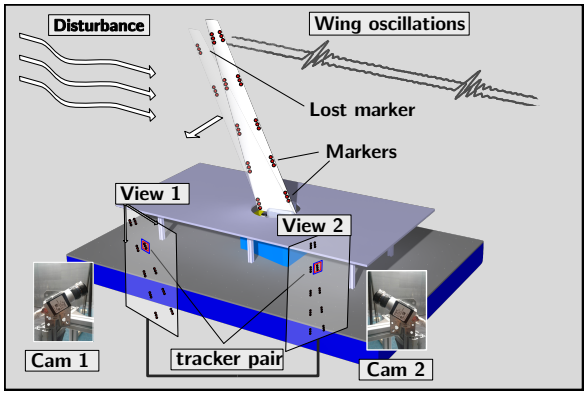


Figure I.1.2: Experimental setup with the wing facing the wind tunnel, equipped with visual markers.

This Chapter is structured as follows. The methodology is presented in Sec. I.1.2, where Sec. I.1.2.2 deals with the segmentation and filtering approach. Two clustering methods, DBSCAN and Disjoint-set data structure, are discussed in Sec. I.1.2.3, with a detailed description of the novel formulation of the DBSCAN, DBSCAN^{-1}, in Sec. I.1.2.4. The experimental setup and the data acquisition process are explained in Sec. I.1.3, with Sections I.1.3.1 to I.1.3.3 covering the setup, hardware and experimental conditions. Furthermore, Sec. I.1.3.4 expands on how the validation dataset was created by the automatic labelling tool specifically designed for this study, while Sec. I.1.3.5 covers the performance test developed for a comparative assessment of the two earlier presented clustering methods. The results of the clustering methods and the full tracking pipeline deployed on the experimental data are discussed in Sec. I.1.4. Finally, the conclusions and recommendations are presented in Sec. I.1.5.

I.1

I.1.2. Methodology

The method proposed in this Chapter describes a computer vision and machine learning approach composed of a robust segmentation-clustering pipeline capable of automatically detecting and extracting marker locations and dealing with temporary marker loss. An image filtering pipeline (segmentation) is implemented consisting of HSV filter and the Otsu's automatic thresholding method [24]. Two machine learning routines are then evaluated: (clustering) DBSCAN [21] and Disjoint-set data structure [22]. The segmentation pipeline is used to extract the point data of the markers, and the clustering is used to label the cluster centroids correctly. The approach was tested on an image sequence of a flexible wing undergoing motion, equipped with active LED markers.

I.1.2.1. Overview of the Complete Tracking Pipeline

A high-level overview of the complete tracking pipeline developed for this study is shown in Fig. I.1.3.

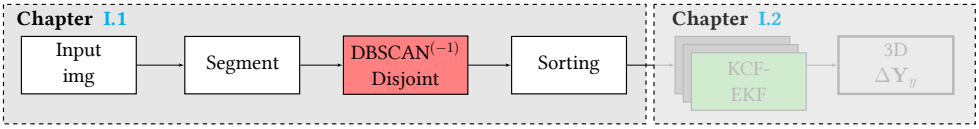


Figure I.1.3: High-level overview of the full tracking pipeline.

The segmentation block refers to the segmentation and HSV filtering processes, addressed in Sec. I.1.2.2. The red block is the clustering algorithm (DBSCAN/DBSCAN^{-1}/Disjoint) implemented in this study as detailed in Sections I.1.2.3 and I.1.2.4. The green blocks represent an independent tracking filter and Kalman Filter duos (KCF-EKF) that run parallel to keep track of the markers through a sequence of images. The output is the marker displacement in $(x,\ y)$ pixel coordinates of the frame. The cyan block is an additional sorting step needed for consistent tracking of the markers, explained in Sec. I.1.2.5. The algorithms presented in this study are mainly concerned with the dotted part, as shown in the schematics in Fig. I.1.3 and aim to highlight the methodology needed to arrive at the inverse DBSCAN (DBSCAN^{-1}) algorithm, the main contribution of this study.

I.1.2.2. Segmentation

Segmentation approaches are generally focused on finding a filter or a sequence of filters $f(I(x,y))$ in order to shape an input image $I(x,y)$ to the desired output $G_f(x,y)$ by altering the pixel intensity values:

$$G_f(x,y) = f(I(x,y)) \qquad I(x,y) \longrightarrow \boxed{f} \longrightarrow G_f(x,y)$$

For a sequence of images, the process is a function of the number of frames and, thus, implicitly, time [26]. When the desired image segments contain colour information, a commonly applied technique is colour filtering in the HSV space. The main benefit of processing in this colour space is that the image intensity and colour can be distinctly separated. The method is also widely used in video sequence processing and image extraction [23, 27].

HSV FILTER

To separate the background from the markers, an HSV filtering pipeline composed of multiple filters is used. First, the image is segmented based on the colour temperature of distinct LED markers, based on distinct values of hue, saturation and value. The filter is tuned to find the near-optimal HSV values to minimise the noise in the image. In Fig. I.1.4, the result is shown of such an operation. From left to right, the figures show how the original

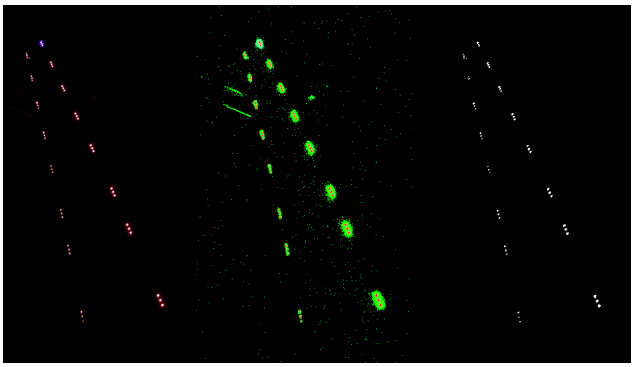


Figure I.1.4: Single HSV filtering operation: original (left), HSV (centre), Black-and-White (binary) (BW) threshold (right) image.

image is filtered based on its HSV values, obtaining a binary Black-and-White (binary) (BW) colour-filtered image. Then, default thresholding is applied to remove the scattered noise from the light diffusion from LEDs and the remaining background. The result is a BW image, a binary mask with distinct LEDs. Hereafter, the contours of the shapes contained in the binary mask are extracted, and the clustering can be applied to identify individual markers. The contours extraction filter is based on the Topological Structural Analysis algorithm of binary images and shapes [28], where a border following technique is applied with topological analysis of the contours of a border shape.

In Fig. I.1.4, the HSV operation is shown when the images are tracked in low lighting conditions. When lighting conditions change, HSV filtering operation may produce a noisy mask, meaning that aside from a distinct mask with LEDs, additional scattered background pixels are present in the HSV (middle) image. Since this image is close to bimodal by nature, it was investigated how the bimodal Otsu's thresholding can improve the segmentation with an additional HSV filtering step based on the image histogram. In Fig. I.1.5, a simplified schematic is shown of the HSV segmentation and clustering pipeline.

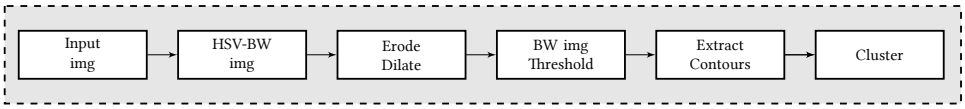


Figure I.1.5: Schematic of HSV filtering and thresholding resulting in BW image needed for the subsequent clustering process.

Morphological Operations

The HSV filter alone may produce a noisy speckle masked image. A typical way to deal with this is using morphological image transformations [29]. Morphological operations are, in general, useful not only for the removal of global noise (e.g. Gaussian noise) but also for isolating and joining separate individual elements. A commonly used cascaded operation is *erode*, followed by *dilate*, where the former *erodes* away pixels and pixel groups captured by a certain kernel size, and the latter *dilates* and enlarges bright pixel groups. Both of these image transformations perform, in essence, a convolution operation of image $I(x, y)$ with kernel $B(x', y')$. Erode operator performs a local min operation with a kernel of the desired size (e.g. 3×3), anchored at the centre. As the kernel slides over the image, the pixel value under the anchor point is replaced by the min value of the region covered by the kernel $B(x', y')$. Dilate operator works according to the same principle but performs a local max operation. The operations can be summarised as follows:

$$f_{erode}(I(x,y)) = \min_{(x',y')\in B_{ker}} (I(x+x', y+y')) \tag{I.1.1}$$

$$f_{dilate}(I(x,y)) = \max_{(x',y')\in B_{ker}} (I(x+x', y+y')) \tag{I.1.2}$$

and combined operation:

$$f_{morph}(I(x,y)) = f_{dilate}(f_{erode}(I(x,y))) \tag{I.1.3}$$

For an appropriate kernel size, this will remove away noisy speckles surrounding and scattered around thresholded shapes. In this study, the kernel size was set to 2×2 pixels. The relevance and effect of morphological operations will be further discussed in Sec. I.1.4.

Thresholding

The thresholding strategy in image processing is essential for obtaining a good mask for DBSCAN clustering. Variations of light and motion activity of the object make the task of obtaining good thresholding for live images challenging [30]. A robust approach has to anticipate the variations in the pixel intensities to produce the best possible mask. Several methods are possible; this study investigated three approaches: global unit normalisation, baseline normalisation, and adaptive global thresholding using Otsu's method [24].

Global Normalisation Thresholding The global normalisation can be applied by converting the 3-channel RGB input image to greyscale. Subsequently, the image can be scaled with the maximum value of the greyscale, depending on how the greyscale is represented (0,1) or (0,255). Then, a single threshold can be applied to obtain a binary mask $G(x, y)$. For an input image $I(x, y)$, this process can be represented as:

$$G(x,y) = f_{norm}(I(x,y)) \tag{I.1.4}$$

$$G(x,y) = \begin{cases} 1, & I(x,y)_{norm} \geq \tau_{th} \\ 0, & I(x,y)_{norm} < \tau_{th} \end{cases} \tag{I.1.5}$$

where the $I(x, y)_{norm}$ can be computed using a simple scaling, or mean μ_I and standard deviation σ_I of the image:

$$I(x,y)_{norm} = \frac{I(x,y) - \mu_I}{\sigma_I} \tag{I.1.6}$$

The downside of this approach is that it does not consider the variations in pixel intensities throughout the image sequence that may have been influenced by changing light conditions and/or movement of the object being tracked. The threshold parameter τ_{th} is, in this case, obtained and tailored for a single static image. The quality of the thresholding then depends on the carefully chosen threshold parameter and predictability of the light variations. When applied correctly to a continuous image sequence, in this particular application, an arbitrary thresholding routine should be able to segment the foreground as a moving object (high intensity) and detect the background as static (low intensity).

Baseline Thresholding In this approach, the baseline pixel intensities are taken into account of the k^{th} image. The first image is a good basis for obtaining a suitable threshold parameter such that variations are taken into account from these baseline values. This process can be represented in a way similar to Eq. I.1.5, but now the normalisation of the k^{th} sequential image (in range $i = 1, 2, \ldots N$) is done according to:

$$I(x,y)_{norm_{i=k}} = \frac{I(x,y)_{i=k} - \mu_{I_{i=k}}}{\sigma_{I_{i=k}}} \frac{1}{I(x,y)_{i=0}} \tag{I.1.7}$$

The downside of this approach is that the sensitivity to the threshold parameter increases and the intensities lie closer together. However, an offset is maintained concerning the baseline in each image sequence.

Adaptive Otsu Thresholding Otsu's automatic global thresholding method tries categorising an image into two classes, background and foreground pixels [24, 31, 32]. The method is well suited for images with a bimodal grey pixel intensity histogram; in this case, the histogram will show two distinct peaks and sharp separation between them, where one peak is assumed to correspond to the bins of the background and the other to the foreground. The threshold value is chosen such that the inter-class variance is minimised, which would suggest placing the threshold value in the middle of the peaks. The minimisation procedure for finding a threshold value of τ_{th} can be represented as:

$$\sigma_w^2(\tau_{th}) = w_1(\tau_{th})\sigma_1^2(\tau_{th}) + w_2(\tau_{th})\sigma_2^2(\tau_{th}) \tag{I.1.8}$$

where the parameters w_1, w_2 and σ_1^2, σ_2^2 correspond to the probability and the variance of the two classes and can be computed from the histograms [24].

The limitation of this method is the bimodality assumption, which may not hold for each image and its greyscale image pair [33]. The histogram may not show clear distinctions when the object is considerably smaller than the surrounding background. Additionally, noise may affect the histogram representation. Variations of Otsu's algorithm exist, which are capable of dealing with noisy images [31]; however, in this regard, HSV filtering is responsible for filtering out most of the image noise, making the thresholding less complicated.

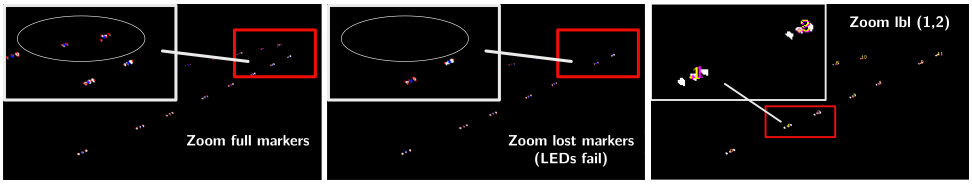
I.1.2.3. CLUSTERING APPROACH

A machine learning approach is used to tackle the problem of correctly detecting and clustering the markers. This study implements and compares two machine learning methods

I.1

for clustering, DBSCAN [21] and the Disjoint-set data structure [22]. These algorithms were particularly suitable due to their unsupervised nature, namely (i) minimum needed domain knowledge, (ii) ability to find clusters of varying size and (iii) ability to deal with noise (in the case of DBSCAN). DBSCAN differs from the Disjoint-set data structure by its ability to deal with noise in the dataset and achieves the goal at a significantly lower computational cost ($O(n \log(n))$). The two unsupervised clustering algorithms are implemented in the marker recognition pipeline and evaluated for speed and robustness performance.

In this study, it was crucial to apply a robust unsupervised clustering method such that an arbitrary number of markers could be accounted for automatically. The robustness assessment was implemented in the experimental conditions, where, due to failure of the LEDs (going on and off), the number of markers (and thus cluster centres) varied over time ánd across experimental runs from a nominal (complete) marker set. Within a single experimental run, the failure was mainly periodic and manifested itself due to high gust loads and wing oscillations. The clustering assessment of an incomplete and complete set of markers is illustrated in Fig. I.1.6. Here, the red dots are contours of point groups found in the clustering mask, and the blue dots are their respective centroids. The image on the right shows the result of clustering.



(a) Complete set of markers.

(b) Successful clustering of an incomplete set of markers.

(c) Comparison DBSCAN scan (purple), and the Disjoint-set data structure (yellow).

Figure I.1.6: The necessity of unsupervised clustering: incomplete (right, middle) versus full (left) set of markers.

DBSCAN

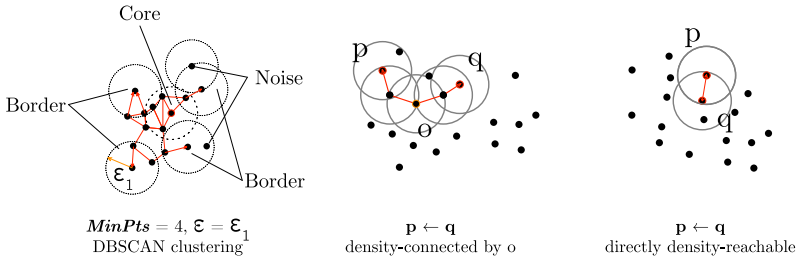


Figure I.1.7: Illustration of the DBSCAN clustering method.

The main principle of DBSCAN is to identify and separate high-density regions from low-density regions. At any given point, p, density is measured within a circular radius ϵ.

A dense region of radius ϵ from point p is a region that contains at least a $MinPts$ number of points; $MinPts$ and ϵ are the main parameters of the algorithm. Given a database D, the ϵ neighbourhood, N_ϵ, of point p w.r.t. point q has the following form [21]:

$$N_\epsilon(p) = \{q \in D | dist(p,q) \leq \epsilon\} \tag{I.1.9}$$

This definition alone, when used naively, will fail to distinguish core points (points inside the cluster), border points (points at the border of a cluster), and noise (a point not belonging to any cluster). The reason is that the ϵ neighbourhood of border points generally has much fewer points than the ϵ neighbourhood of a core point.

The problem arises when the $MinPts$ parameter is set to a low value to include the border points, which can also cause noise to be included in the cluster. To overcome this, DBSCAN introduces the concept of density reachability. A point is said to be *Directly Density Reachable* when the following two conditions hold:

$$p \in N_\epsilon(q) \tag{I.1.10}$$

$$|N_\epsilon(p)| \geq MinPts \quad \text{(core point condition)} \tag{I.1.11}$$

These conditions, thus, set a requirement for every point p in a cluster to be in the ϵ neighbourhood of another point q in this cluster. Additionally, the ϵ neighbourhood of q, $N_\epsilon(q)$, must have a minimum of $MinPts$, classifying it as a core point. The method further introduces connectivity conditions for connecting N_ϵ of points and defines noise as a point not belonging to any cluster in dataset D under the given conditions (Density-Reachability and connectivity) [21]. The basis of the clustering approach and the definitions are illustrated in Fig. I.1.7. As shown, point p can be reachable from point q by Density-Reachability or connectivity.

DISJOINT-SET DATA STRUCTURE

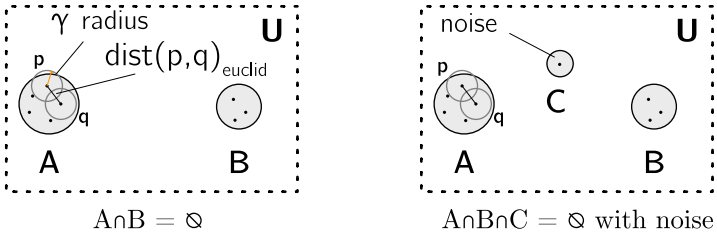


Figure I.1.8: Venn diagram and illustration of Disjoint-set clustering method.

Disjoint-set data structure operates by organising a set of elements into a distinct number of disjoint sets, also referred to as equivalence classes [22]. For a given data set D, obtained as a result of filtering and contour operations, equivalence classes are defined that are non-overlapping. Subsets A and B are considered a disjoint-set when the overlap U between them belongs to an empty set $\varnothing$:

$$A \cap B = \varnothing \tag{I.1.12}$$

The algorithm assigns *all* points of the dataset to an equivalence class hence no inherent mechanism is built-in to cope with noise, and a noise particle may belong to a dedicated subset C. Consequently, and as will become more evident in the following sections, a good filtering approach is needed with this method to remove the noise.

To make the method comparable to DBSCAN, the threshold for the disjoint-sets can be defined with a distance metric, radius γ, similar to ϵ. A set of points,$\{p, q, \ldots\}$ belongs to a disjoint-set A, when they are packed within radius tolerance γ, resulting in the following conditions:

$$A = \{p, q, \ldots\} \tag{I.1.13}$$

$$dist\,(p, q)_{euclid} \leq \gamma \tag{I.1.14}$$

Here, the latter condition is defined as the euclidean norm of points p and q:

$$dist\,(p, q)_{euclid} = \sqrt{(p(x, y) - q(x, y))^2} \tag{I.1.15}$$

The presented definitions are illustrated in Fig. I.1.8. In Fig. I.1.6c a comparison is shown of the clustering operations for DBSCAN scan (purple) and Disjoint-set data structure (yellow).

I.1.2.4. Inverse DBSCAN: DBSCAN^{-1} — a Novel Clustering Approach for Sparse Datasets

While DBSCAN allows explicit definition for noise in the data (points not meeting the core points condition), the success in rejecting the noise is closely tied to the correct selection of parameters and the quality of the thresholded input image. The clustering becomes harder when high-density noise is introduced into the data. Noise can have various sources, e.g. interference in hardware signal, poor illumination or, simply, poor pre-filtering and thresholding of the input image. There are also conditions where pre-filtering, such as the morphological operations, is not possible or has adverse effects (further elaboration follows in Sec. I.1.4). In particular, for sparse datasets, under such conditions, DBSCAN is known to fail to identify the desired clusters [34]. This shortcoming arises from the fact that for a high density of scattered noise, noise particles are more likely to meet the core point criteria for a given DBSCAN parameter set.

To remedy this problem, a novel formulation of DBSCAN is proposed, the inverse DBSCAN, denoted by DBSCAN^{-1}. In this new model, a different perspective on the clustering problem is needed: instead of trying to *reject* the noise, it is proposed to *actively look* for noise. Hence, DBSCAN^{-1} tries to detect noise explicitly, and clustering becomes an implicit task. The proposed approach would be to utilise this formulation of DBSCAN as a noise removal filter, then apply nominal DBSCAN again on the clean image domain. To enable this approach, a redefinition of DBSCAN is needed. For a given database D, the ϵ neighbourhood of noise particles p_n and q_n is defined as:

$$Z_\epsilon\,(p_n) = \{q_n \in D \mid dist\,(p_n, q_n) \leq \epsilon\} \tag{I.1.16}$$

DBSCAN, in its original form, was intended for obtaining clusters for large datasets and relatively low noise; hence no limitation is set on the maximum number of clusters. In the

definition of DBSCAN^{-1}, an additional parameter, denoted by $MaxPts$, is introduced, which sets a cap on the allowable number of points in the ϵ neighbourhood of noise p_n, denoted by Z_ϵ. The noise particle is directly reachable from another cluster of noise particle(s) when the following holds:

$$p_n \in Z_\epsilon\left(q_n\right) \tag{I.1.17}$$

$$MaxPts \geq \left|Z_\epsilon\left(p_n\right)\right| \geq MinPts \quad \text{(core noise particle condition)} \tag{I.1.18}$$

Three conditions must be placed on the DBSCAN^{-1}: (i) $MinPts$ must be set to 1 to capture individual noise particles; (ii) ϵ must be at least the standard deviation of the noise density, σ_n (σ_{x_n} and σ_{y_n}) in the spatial domain in terms of $(x,\, y)$ coordinates for zero mean distribution; and (iii) $MaxPts$ must count *less* points than ϵ neighbourhood of desired cluster points, $N_\epsilon\left(q\right)$, a condition that is directly related to the standard deviation, $\sigma_{cluster}$ ($\sigma_{x_{cluster}}$ and $\sigma_{y_{cluster}}$) of $(x,\, y)$ coordinates of a dense cluster and can be chosen based on a priori analysis of the input dataset. These conditions dictate that point noise particle p_n does not belong to the ϵ neighbourhood of true clusters N_ϵ, but to Z_ϵ:

$$\begin{cases} p_n \in Z_\epsilon \\ p_n \notin N_\epsilon \end{cases} \qquad \text{and} \qquad \begin{cases} MinPts = 1 \\ MaxPts < \sqrt{(\sigma_{x_{cluster}} + \sigma_{y_{cluster}})} \\ \epsilon \geq \sqrt{(\sigma_{x_n} + \sigma_{y_n})} \end{cases}$$

density reachability parameter constraints

A necessary condition for this is that if a probability distribution of points is defined on the 2D image plane in dataset D as $P(x,y) = \iint_D$, the density distribution of the desired particles, $P(x,y)_{cluster}$, is higher than the density distribution of the noise, $P_n(x,y)$; otherwise, the true clusters will dissolve in the noise:

$$P(x,y)_n < P(x,y)_{cluster} \tag{I.1.19}$$

If the above condition is not met, the clustering will fail for the given condition of dataset D. What this clustering model will do, in essence, is detect the group of desired clusters as points surrounded by *too many other points* (filtered by the max $MaxPts$ conditions) and reject them as noise. The actual noise particles will meet the core noise particle condition of DBSCAN^{-1} as they lack a distinctive concentrated distribution.

A visual representation of this process and the relevance of the DBSCAN^{-1} — in particular in the absence of morphological operations — can be found in Sec. I.1.4.

I.1.2.5. RADIAL SORTING

Obtaining the cluster centre locations in the frame after the clustering routine provides only a static map of the markers without a spatial orientation concerning the underlying geometry. A radial sorting algorithm is proposed in the processing routine to obtain a geometrical representation behind the detected clusters. This algorithm represents the cyan block in Fig. I.1.3. The algorithm is initiated by finding the centroid $\bar{\mathbf{c}}_{cp}$ of the cluster centres (a cloud of points) $\mathbf{P}(x,y)$, then obtaining a radially sorted distribution, a so-called

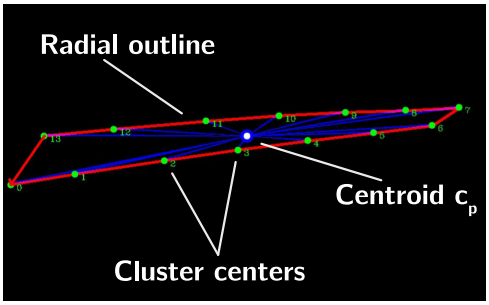


Figure I.1.9: Radial sorting algorithm process. The cluster centres are green dots; the convex radial hull is the red outline.

convex radial hull, $\mathbf{P}_{\theta hull}$, of n indices, such that the outline of the hull has a continuous connectivity.

First, the algorithm takes as input an arbitrarily indexed cloud of cluster centres, denoted $\mathbf{P}(x, y) \in \mathbb{R}^{2\times n}$. The centroid of $\mathbf{P}$, $\bar{\mathbf{c}}_{cp}(x, y)$ is calculated to obtain the vector pointing towards the centroid. If the input is a continuous shape, the centroid is sampled at the contours of the area; otherwise, for a collection of n points:

$$\bar{\mathbf{c}}_{cp} = \frac{1}{n}\sum_{i=1}^{n} p_i \text{ and,} \tag{I.1.20}$$

$$\mathbf{d}_{cp} = \mathbf{P} - \bar{\mathbf{c}}_{cp} \tag{I.1.21}$$

Next, the angle defined by the direction of each vector is calculated, and the resulting vector of angles is radially sorted around $\bar{\mathbf{c}}_{cp}$ in the given orientation to obtain the convex radial hull $\mathbf{P}_{\theta hull}$:

$$\boldsymbol{\theta}_{cp} = \text{ARCTAN2}\left(\mathbf{d}_{cp}\right), \text{ where for each point,} \tag{I.1.22}$$

$$\text{ARCTAN2}\left(p\right)_i = \text{ARCTAN2}\left(\frac{p_y}{p_x}\right)_i \tag{I.1.23}$$

Where, p_x and p_y indicate the pixel locations in x and y, respectively. Then, the sorted index of angles is obtained from $sort(\boldsymbol{\theta}_{cp})$, and the convex radial hull is obtained from sampling by this sorted index:

$$\mathbf{P}_{\theta hull} = sort\left(\mathbf{P}, sort\left(\boldsymbol{\theta}_{cp}\right)\right) \tag{I.1.24}$$

This is required to outline points in a continuously connected area. The process of radial hull sorting is shown in Fig. I.1.9.

Algorithms such as Jarvis march [35] also use a form of radial sorting to wrap a cloud of points in a convex hull. The main difference with the Jarvis march is that the radial sorting algorithm is intended for obtaining a continuous geometry by sorting *all* cluster centres through a continuously connected outline. With a convex hull, some cluster centres may fall inside the convex hull region and hence be excluded from the outline. The

other difference is that Jarvis march is done at a complexity of $O(nh)$ (n points and h hull corners), while in the proposed approach, the sorting can be done in one pass at $O(n)$ complexity.

I.1.2.6. Reconstruction

The reconstruction is the final step that relates the displacements of corresponding markers in two frames and reconstructs the 3D displacement. The reconstruction process can be inferred from the schematic of the camera setup shown in Fig. I.2.5 shown in Chapter I.2. Further details regarding the 3D reconstruction can be found in a previous study by Mkhoyan et al. [16], and [36].

I.1.3. Experimental Setup and Data Collection

The experimental data was collected from camera observations of a flexible wing undergoing gust excitations equipped with active LED markers. This experiment was performed within the scope of a larger study on smart sensing methods for controlling flexible aircraft.

I.1.3.1. Apparatus

The experiment was conducted in the Open Jet Facility (OJF) at the Delft University of Technology [37]. The OJF, as shown in Fig. I.1.10, is a closed-circuit low-speed wind tunnel, driven by a 500 KW electric engine, with an octagonal test section of 285×285 cm^2. The maximum flow velocity in the wind tunnel is 35 m/s; however, the theoretical performance limit is around 30 m/s.

A gust generator composed of two servo-controlled foam wings was installed in the test section to facilitate various dynamic motion conditions during the test. This particular gust generator allows gust vane deflections of $|\alpha_g| \leq 15°$, or $10°$, depending on the actuation frequency (5-7 Hz or 10-15 Hz), and can produce a harmonic signal, as well as sweep signals of varying frequencies.

A Polytec PSV-500 laser vibrometer system [38] with a resolution (RMS) of 200 µm/s was used to measure the dynamic response of the wing to the aerodynamic loads introduced by the gust onsets. The PSV system was configured to measure 8 (active) markers, as shown in Fig. I.1.12a from a total of 16 LED markers placed on the wing. The numbering of the marker ID's in the image tracking algorithm is indicated with square braces and the laser tracking system in square braces. Since the laser allowed for the measurement of only a single point for each run, each run would be repeated eight times to reconstruct the displacement field of the wing. The system was configured for a sampling rate of 400 Hz.

As shown in Fig. I.1.10 and, schematically, in Fig. I.1.2, a pair of cameras were used to observe the motion of the wing. These cameras are referred to as leading-edge camera (Cam 1) and trailing-edge camera (Cam 2), respectively.

I.1.3.2. Wing Model and Motion Conditions

The wing used in the experiment, referred to as the *Allegra wing*, is a forward-swept tapered wing built of glass fibre reinforced plastic. The design of the wing allows for large tip

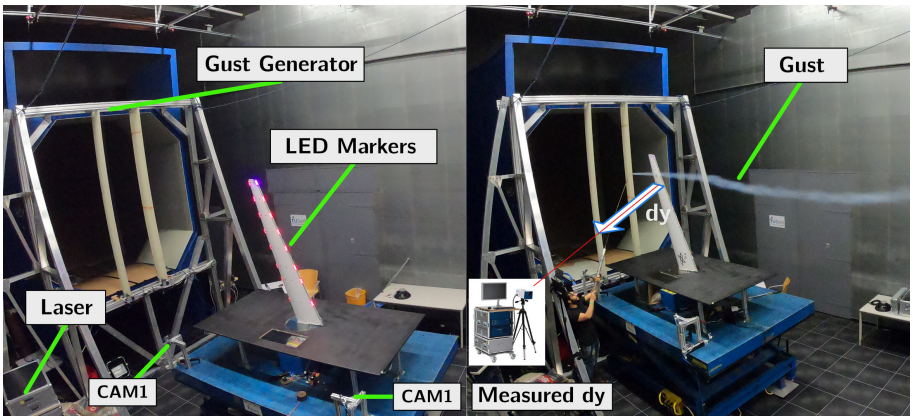


Figure I.1.10: The Open Jet Facility (OJF), showing the gust generator mounted in front of the test section.

displacements, up to 20% for 10° of Angle of Attack (AoA) and 50 m/s flow velocity [39]. The wing was clamped on one side on a sturdy table under a fixed angle of attack of 4°. Detailed information about the wing can be found in Appendix I.A1.1.

The wing was equipped with 16 LED markers. Each LED marker consisted of 3 sub-LED units, providing three distinct bright light sources per marker. In the experiment, a 1-cos gust signal and a frequency sweep signal were used.

The data collected for three experimental conditions or runs denoted R1, R2 and R3, as listed in Table I.1.2. The experimental variables are the flow velocity in the wind tunnel, V_{INF}, the gust vane frequency, f_g, and gust vane angle α_g.

Table I.1.2: Flow and motion conditions for runs R1, R2 (discrete gusts) and R3 (sweep).

Run ID	Frequency [Hz]	Vane angle [°]	Flow velocity [m/s]	N images [-]	N gusts [-]
R1	5	10	30	469	3
R2	5	5	30	469	3
R3	0.1 - 10	5	30	574	-

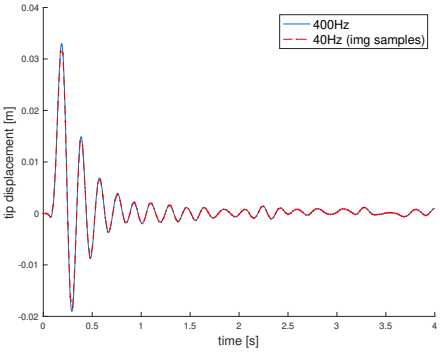
For all runs, the images were first recorded in *dark* conditions (night visibility), meaning the lighting conditions were low for good visibility of the LEDs. Additionally, *bright* images (daylight visibility) were collected to study the effect of HSV filtering in high visibility conditions.

The gust generator parameters were selected such that the disturbance produced a high dynamic response from the wing to ensure sufficient pixel activity in the image. The gust vane frequency of 5 Hz was close to the wing's natural frequency at the given mass configuration. Runs R1 and R2 each contained three consecutive gust inputs; run R3 did not have a discrete gust but a sweep signal. The purpose of the R2 run was to act as a control against the results of R1, while R3 was designed to show marker loss (LEDs on/off) under high dynamic activity.

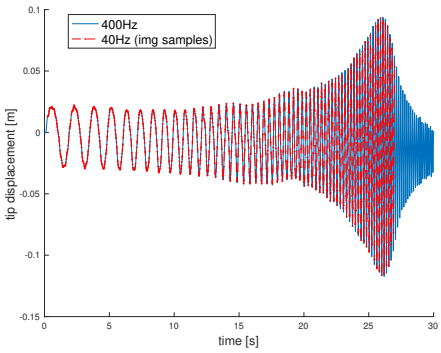
I.1.3.3. Dataset Collection

Measured Wing Response

The time history signals in Fig. I.1.11 correspond to the measurements taken at the location of marker ID 1; the labelling of the marker IDs for the vibrometer measurement system can be found in Fig. I.1.12a. Figure I.1.11a shows the wing's response to a single gust input; Fig. I.1.11b shows the response to a sweep signal. The blue curves correspond to the measurements taken by the laser vibrometer sampled at 400 Hz; the red curves are spline models of this response sampled at the capture intervals by the leading edge camera. The spline model is required to obtain synchronised measurement points between the laser vibrometer data and the image sequences for comparison. The camera images were collected at approximately 40 Hz, with the Nyquist frequency well above the expected resonance frequency of the wing of ≈ 5 Hz.



(a) Gust signal (1-cos): $\alpha_g = 10°$, $f_g = 5$ Hz, $V = 30$ m/s.



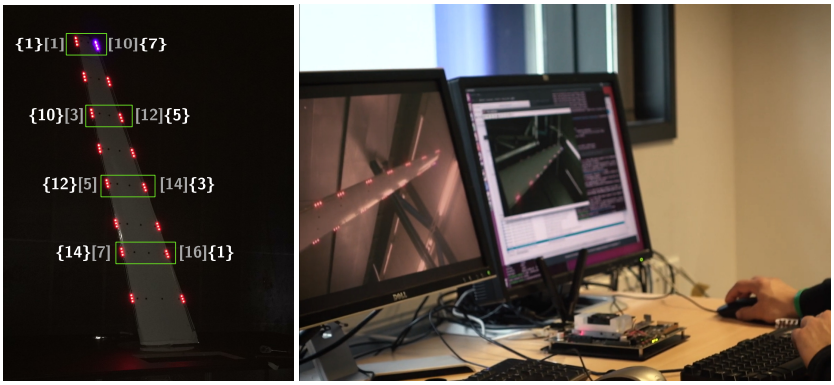
(b) Sweep signal: $\alpha_g = 5°$, $f_g = 0.1 - 10$ Hz, $V = 30$ m/s.

Figure I.1.11: Laser vibrometer measurement (blue line), of the tip displacement of marker ID 1, sampled at capture intervals of Cam 1 at ≈ 40 Hz (red line); runs with discrete gust (left) and sweep signal (right).

Hardware Setup

An overview of the data acquisition hardware is shown in Fig. I.1.12. The dataset was recorded with two GigE acA1300-75gc Ethernet Basler cameras with 1300 CMOS 1.3 megapixel (1280×1024 pixels) sensor [40]. The cameras were equipped with Computar 12 mm F1.4 2/3" P IRIS lenses [41] and were positioned in a stereo setup to observe the markers from two viewpoints. The resulting image was cropped to 1088×600 pixels and streamed in 3 channel RGB format synchronously via real-time PTP triggering protocol over the Ethernet. A Power over Ethernet (PoE) smart switch GS110TP from NETGEAR provided both the power, 3.5 W (per camera unit), as well as the GigE capability to stream the images up to 140 Frames Per Second (FPS).

An embedded computing system delivered the processing power and image capture during the experiment from NVIDIA, the Jetson TX2, equipped with NVIDIA Pascal architecture with 256 NVIDIA CUDA cores and 1.3 TFLOPS (FP16), Dual-core Denver 2 64-bit CPU and quad-core ARM A57 complex [42]. The Jetson TX2 is designed for embedded applications using Artificial Intelligence (AI) and Computer Vision (CV) and operates



(a) The actively measured markers. (b) Data acquisition and live tracking hardware setup.

Figure I.1.12: The experimental setup, showing the active markers measured by the Polytec measurement system (in green) and the Jetson TX2 hardware setup.

on Ubuntu 16.04 LTS allowing flexibility in code deployment. The application developed for this study was programmed in C++ and deployed on the device. For the development the algorithms Basler C++ Pylon API [40] and the OpenCV open-source computer vision library [43] were implemented. To perform image segmentation, capturing and compression, GPU hardware acceleration [44] was used with Jetson TX2 dedicated GStreamer pipelines [45].

Code development, algorithm testing and assessment were done using CPU processing with a standard Dell Optiplex 7400, a 2.3 GHz Intel Core i5 16G MacBook and the Jetson TX2. The image and tracking data were extracted and plotted using the OpenCV-Matlab parsing interface tmkhoyan/cvyamlParser [46]. The code, dataset and tools developed are available under the repository tmkhoyan/adaptiveClusteringTracker [47].

I.1.3.4. Validation Dataset

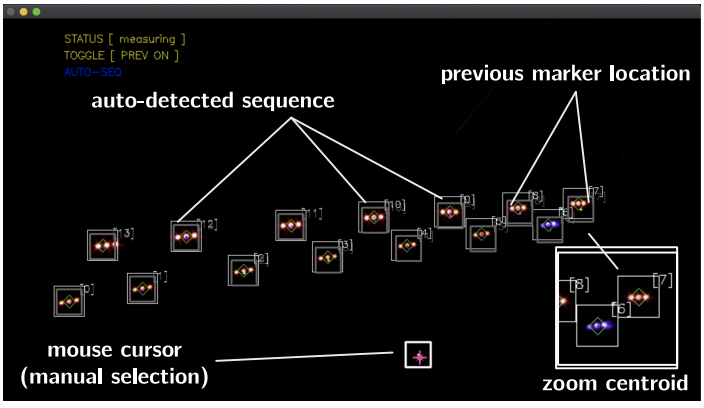


Figure I.1.13: The labelling process with the automatic labelling tool.

An automatic labelling tool was developed and implemented to create a reference dataset of the image sequence R1 from Table I.1.2 in order to perform a comparative assessment of the two clustering methods DBSCAN and Disjoint-set data structure. The tool allows automatic tracking and labelling of the pixel-wise (x, y) location of the markers through a sequence of dynamic images, given an initial hand-labelled marker-set in the first image. The capability to track a sequence of images classifies it as a tracking routine. However, each consequent frame is visually checked before the labelled data is saved to ensure the reference dataset's validity.

The processing strategy of the tool can be summarised in the following way:

- Manually select initial marker locations of the first image in the sequence. The marker locations are defined as a (x, y) pixel location of the centre LED of each 3-LED marker cluster.
- A sub-matrix is defined as a bounding square box enclosing the 3-LED cluster at the $(x,\ y)$ location of the centre LED, with a width of 40 pixels.
- The sub-matrices corresponding to the number of markers (14 in total for the R1 image sequence) serves as an input to the automatic detection of the markers in the next image frame.
- A detector is implemented to process each sub-matrix, defined at the location of the sub-matrix from the previous image with an uncertainty factor of 1.2 in width and height (i.e. the bounding square is factor 1.2 larger than the initial sub-matrix).
- Each automatic detection is visually approved before moving to the next frame and saving the data.

The uncertainty margin (1.2 factor in width and height) is implemented such that the new, shifted marker location can be found with respect to the previous image, and enough margin is kept to account for the motion. This process is depicted in Fig. I.1.13.

The tool enables the implementation of a custom detector for the detection of the circular shaped LED-markers. In the current study, a contour filter, often referred to as a blob detector, was used based on the Topological Structural Analysis algorithm of binary images and shapes [28]. Prior to the detection, the sub-matrix thresholding is applied using Otsu's adaptive thresholding method, such that a binary mask of the marker outline is obtained. This tool was developed in C++ programming language using the OpenCV open-source computer vision library [43], and made available under the BSD-3 licence [48].

I.1.3.5. Clustering Performance Test

A performance test was designed to compare DBSCAN to the Disjoint-set data structure. In this test, a grid, $I(x, y)_{grid}$ of 10000×10000 pixels was used, and clusters of particles were generated randomly to perform the clustering. For each run, the grid was initialised with a varying number of cluster centres, m_i (e.g. 10, 50, 100), with a uniform distribution. The grid size is used as the minimum and maximum bounds of this distribution, with a 0.9 shrink factor to keep 10% free at the borders. The cluster centre distribution is defined as follows:

$$P_{centre} \in I(x, y)_{grid}, \text{ and } \begin{cases} x_{min} & = 0.1 \cdot w_{img},\ x_{max} = 0.9 \cdot w_{img} \\ y_{min} & = 0.1 \cdot h_{img},\ y_{max} = 0.9 \cdot h_{img} \end{cases} \tag{I.1.25}$$

I.1

Around these m_i cluster centres, a fixed number of $n_i = 50$ scatter points was sampled with a normal distribution with the following properties in both x and y locations:

$$P_{cluster} \in I(x,y)_{grid}, \text{ and } \begin{cases} \mu_{cluster} & = \mu_{cluster_x} = \mu_{cluster_y} = 0 \\ \sigma_{cluster} & = \sigma_{cluster_x} = \sigma_{cluster_y} = h_{img}/100 \end{cases} \tag{I.1.26}$$

Here, the sampled normal distribution, $P_{cluster}$, is the offset from a cluster centre (x, y) pixel coordinate; $\mu_{cluster}$ and $\sigma_{cluster}$ are the mean and standard deviation of the distribution. The resulting scatter model is a cloud of points, with the majority falling inside a radius of $\sigma_{cluster}$ (defined as a factor of image width, h_{img}) from the cluster centre). Figure I.1.14 shows a randomly sampled dataset with $m_i = 10$ number of cluster centres and cluster size of $n_i = 50$. In order to assess the clustering methods on their ability to cope with noise, for each run, uniformly distributed noise was generated on top of the existing points. These scattered noise points, n_{inoise}, were proportional to the number of cluster centres with a factor 5 (i.e. $n_{inoise} = m_i \times 5$).

The performance test was used to generate the performance dataset; the code was developed in C++ programming language using the OpenCV open-source computer vision library [43] and made available under the MIT licence [49].

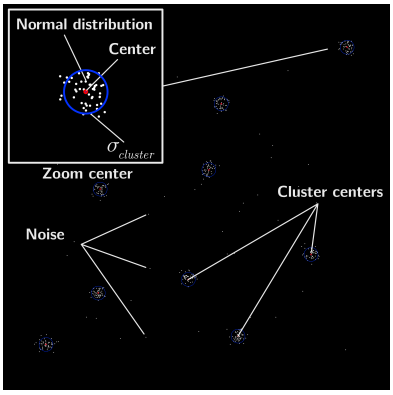


Figure I.1.14: Randomly sampled scatter data for 10 cluster centres and a cluster size of 50.

I.1.3.6. Noise Model

A common noise model was used to evaluate the clustering methods' real-life performance. The input images were injected with an image-independent Gaussian noise, and the robustness of the colour filtering, thresholding and clustering pipeline was investigated against possible sensor noise, transmission and hardware-related issues and poor illumination. Subsequently, the tracking quality of the pipeline was assessed on image sequences from R1 and R3, whereas R2 was used as a reference. The probability density function of the Gaussian noise model is as follows:

$$I(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(z-\mu)^2}{2\sigma^2}} \tag{I.1.27}$$

In this model, z represents the greyscale value. The parameters used for the noise model are a mean of $\mu = 0$ and a standard deviation of $\sigma = 0.5$. The grey values produced from the probability distribution are scaled to RGB range 0-255 and injected in the 3-channels of the input image $I(x, y)$, producing a new additive noise input image $J(x, y)$. The random seed is initialised with the CPU clock (time t) for each image input, $I(x, y)_k$, resulting in a dynamic noisy image input sequence, $I(x, y)_{noise_k}$, at each k^{th} frame:

$$I(x, y)_{noise_k} = I(x, y)_k + N(x, y, t) \tag{I.1.28}$$

Here, $N(x, y, t)$ is the random seed initialised noise mask.

I.1.4. Results and Discussion

The experimental data were processed with two clustering pipelines implementing DBSCAN and Disjoint-set data structure clustering methods. Here, a distinction must be made between the randomly generated *performance dataset* generated with the performance test discussed in Sec. I.1.3.5 and the *experimental dataset*, as provided in Table I.1.2.

The performance test was done to extract the isolated clusters of points from scattered data. Here, the novel implementation of DBSCAN was used, with the additional $MaxPts$ parameter set to 100, $MinPts$ to 20 and ϵ to 180 pix. The ϵ parameter was chosen to have a value approximately twice as significant as the standard deviation of the cluster population, $\sigma_{cluster}$, to capture the majority of the randomly generated cluster points scattered around the cluster centres. For the Disjoint-set data structure, the distance parameter γ was chosen to be equal to ϵ.

Furthermore, the tracking result with the full clustering pipeline on the runs R1 and R3 were performed on a sequence of ≈ 469 images from Cam 1 (Leading-edge). R2 was used as the control for R1 and showed a similar result. R3 was mainly used to assess the ability of the tracking pipeline to deal with marker loss. In the nominal runs, $\epsilon = 20$ pix and the reachability parameter $MinPts = 2$ were used. This set of parameters provided the best cluster detection considering preceding segmentation filters. Here, as in the previous case, the distance parameter γ for the Disjoint-set data structure was chosen equal to ϵ of the DBSCAN.

Speeds of 250+ fps were measured for the DBSCAN implementation on an image sequence of a single camera with a resolution of 1088×600 pixels using a standard Dell Optiplex 7400, a 2.3 GHz Intel Core i5 16G MacBook and the Jetson TX2. The outcomes of both methods were compared to the reference data collected by the automatic labelling tool developed explicitly for this purpose, as addressed in Sec. I.1.3.4.

I.1.4.1. Performance Test of DBSCAN and Disjoint-Set Data Structure Clustering Methods

The performance test was executed with cluster centre sizes $m_i = 5, 10, 50, 100, 200, 500$ and cluster population sizes of $n_i = 50, 50, 50, 50, 50, 50$. For cluster centre sizes $m_i < 100$, the uniform distribution of the cluster centres was balanced to ensure a minimum distance from each cluster centre. This was done in order to prevent cluster populations from merging. For larger cluster centre sizes (≥ 200) merging was allowed.

The results of the clustering are shown, from left to right, for cluster centre sizes $m_i = 10, 100, 200$ in Fig. I.1.15. The purple and yellow radii and their respective centres represent the detected clusters and correspond to the ϵ and γ of the DBSCAN and Disjoint-set data structure, respectively. For all runs, the advantage of DBSCAN with regard to noise is evident. Even in the presence of relatively low noise ($n_{i_{noise}} = 50$ for $m_i = 10$), the Disjoint-set data structure fails and, aside from real clusters, also classifies these noise particles as clusters. DBSCAN, on the other hand, can make this distinction and extract the correct number of isolated clusters. As the population density increases, the initial scatter distribution is not balanced, and certain clusters merge; therefore, the number of detected clusters does not have to correspond to the number of initial clusters. For the remaining two runs, DBSCAN is consistent in performing the task and is able to separate and correctly identify the isolated clusters.

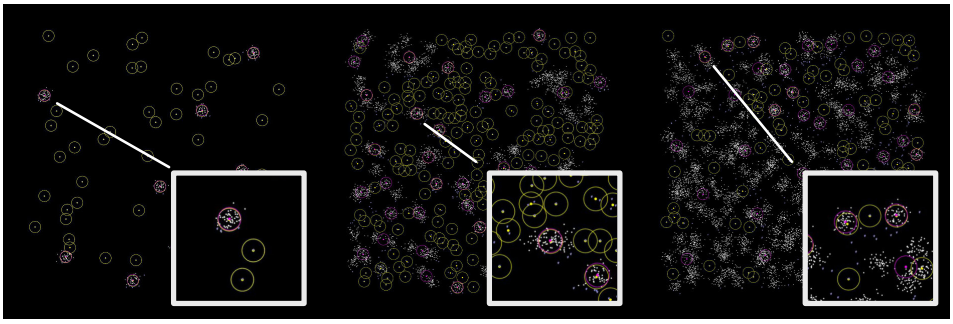


Figure I.1.15: Results of the performance for (left to right) $m_i = 10, 100, 200$ and $n_i = 50, 50, 50$.

The advantage of the novel DBSCAN formulation and the $MaxPts$ was also evaluated for this particular task. With the addition of the $MaxPts$, DBSCAN is able to ignore widely scattered clusters and focuses more on isolated islands of clusters.

I.1.4.2. Validation of DBSCAN and Disjoint-set Data Structure Clustering Pipeline

Two clustering methods were compared and validated against the validation dataset created with the automatic labelling tool. The clustering methods were implemented in the entire tracking pipeline, and the image sequence from the experimental dataset R1 was processed. The images were sampled at the sampling intervals (red line), as shown in Fig. I.1.11. Here the laser vibrometer measurement (in blue), showing the wing's motion, is later compared to the output of the validated tracking pipeline. Sequence R2 showed a similar response to R1 and is therefore not included in the following validation plots.

Spatial Marker Scatter Validation

Figure I.1.16 shows an overview of the clustering results. Here, the marker location (defined as the centroid of the 3-LED contour) in the first image is indicated with yellow, purple and blue circles corresponding to DBSCAN, Disjoint-set data structure and validation data, respectively. The scattered points reflect the marker positions detected in the

complete sequence of R1, where the colour is kept consistent for the data. The motion of the wing, more specifically the tip deflection, reflects in the spread of the scatter points observed. As expected, the spread is the highest for the markers closer to the tip.

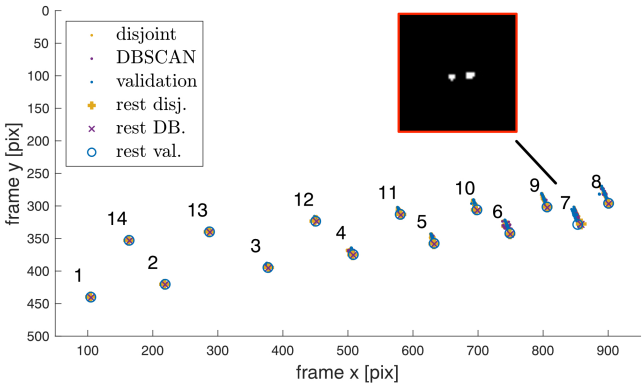
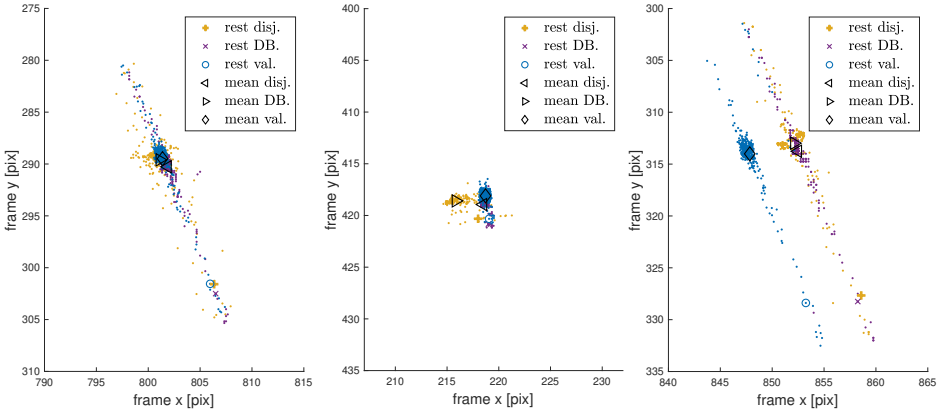


Figure I.1.16: Spread of tracked markers for run R1 with DBSCAN and Disjoint-set data structure.

In the two boxes in Fig. I.1.17, a zoom is shown for maker 9 with the lowest error and markers 2 and 7 with the highest errors concerning the validation data. In these figures, the marker location of the initial image at rest is indicated by a purple cross and a yellow plus sign for DBSCAN and Disjoint-set data structure, respectively, to distinguish the results of the two methods. The diamond and the triangle shapes indicate the mean. Fig. I.1.17a shows a relatively low error, and the spread is closely packed. In Figures I.1.17b and I.1.17c , both methods show a larger spread in (x, y) with respect to the validation data. However, the Disjoint-set data structure has a higher spread, mainly in the x-direction.



(a) Zoom of marker ID 9 (low error). (b) Zoom of marker 2 (high error) (c) Zoom of marker ID 7 (high error).

Figure I.1.17: Zoom spread markers IDs 2 and 7 of run R1 with DBSCAN and Disjoint-set data structure.

To quantify the error, a squared distance error metric is used close to the Root Mean Square Error (RMSE) formulation. The error is defined as the squared average of the Eu-

clidean distance throughout the sequence i to n:

$$RMSE = \sqrt{\sum_{i=1}^{n} dist\left(p, \hat{p}\right)_{euclid_i}} \tag{I.1.29}$$

where the Euclidean norm of the reference point, $p(x, y)$, and its estimate, $\hat{p}(x, y)$, throughout the sequence i to n yields:

$$dist\left(p, \hat{p}\right)_{euclid_i} = \sqrt{\sum_{i=1}^{n} \left(p(x, y)_i - \hat{p}(x, y)_i\right)^2} \tag{I.1.30}$$

From the boxplot in Fig. I.1.18, a better insight can be gained in the average error of the Euclidean distance norm in x and y (Eq. I.1.29). Here, the colour codes are consistent with the clustering methods; furthermore, the diamonds indicate the mean of the data, and the red crosses the outliers. The outliers are defined as the points that are factor 1.5 larger than the bounds of the interquartile range (i.e. data between 25$^{\text{th}}$ and 75$^{\text{th}}$ percentiles). It is observed that the average error through the complete R1 sequence lies below 1 pixel for most of the markers with DBSCAN. Disjoint-set data structure shows a relatively higher spread and a larger mean error. This is observed in particular for markers 1, 2 and 7. For the latter marker, DBSCAN also shows significantly larger errors of up to 4.5 pixels; however, in markers 1 and 2, the Disjoint-set data structure has a factor 2 and 3 larger errors, respectively. These observations are consistent with the results shown in Fig. I.1.17. The large errors can be attributed to the fact that the contours of the 3-LED marker cluster merge due to the wing's motion, and a slight change in the LED-reflection results in 2 distinct contour shapes. Therefore, the centroid of these 3-LED markers falls approximately 3-4 pixels away from the true centroid. This is visible in the close-up box in Fig. I.1.16.

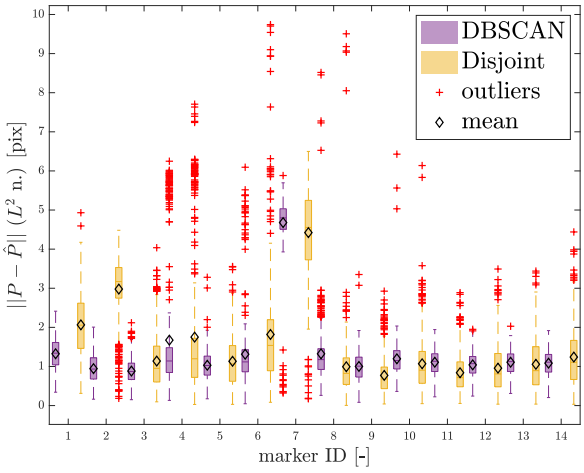
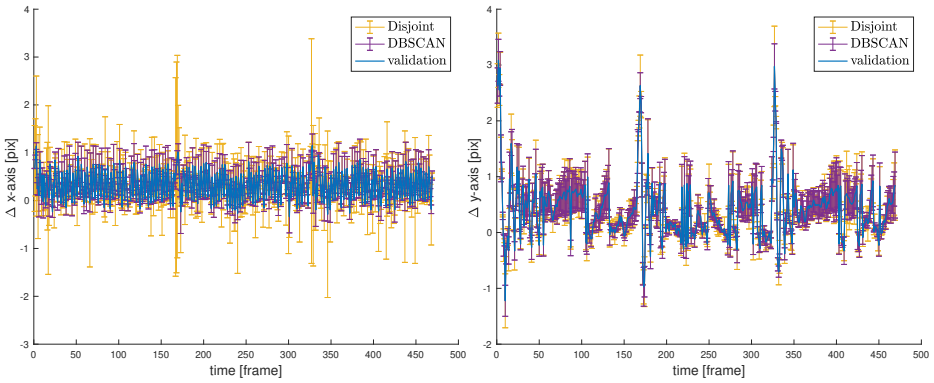


Figure I.1.18: Euclidean norm tracking error for run R1 with DBSCAN and Disjoint-set data structure.

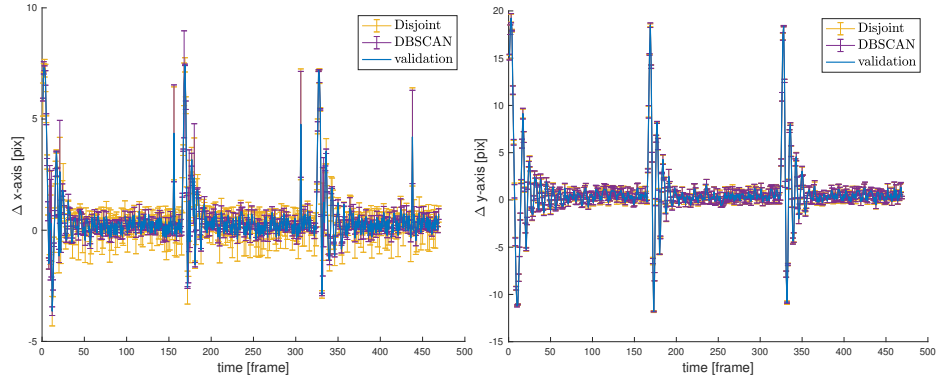
Sequence Tracking Validation

By observing the tracking of markers 2 and 7 across the R1 sequence, one can confirm the larger errors of the Disjoint-set data structure. Figure I.1.19 shows the time traces of the displacement of the markers 2 and 7 in x and y directions, respectively, concerning the steady-state position (initial image). In R1, exactly three gusts were introduced, which can be observed by the three peaks followed by decaying sinusoidal responses. Alongside the displacement, error bars are shown, defined according to Eq. I.1.29. Again, the colour codes are kept consistent, and the Disjoint-set data structure shows a larger error band (up to 3 pixels) compared to DBSCAN and the reference data. In particular, the x-direction exhibits a higher sensitivity to errors, as observed in the presented scatter plots. DBSCAN shows better agreement with the validation data.



(a) Displacement in x of R1 sequence marker ID 2.

(b) Displacement in y of R1 sequence marker ID 2.



(c) Displacement in x of R1 sequence marker ID 7.

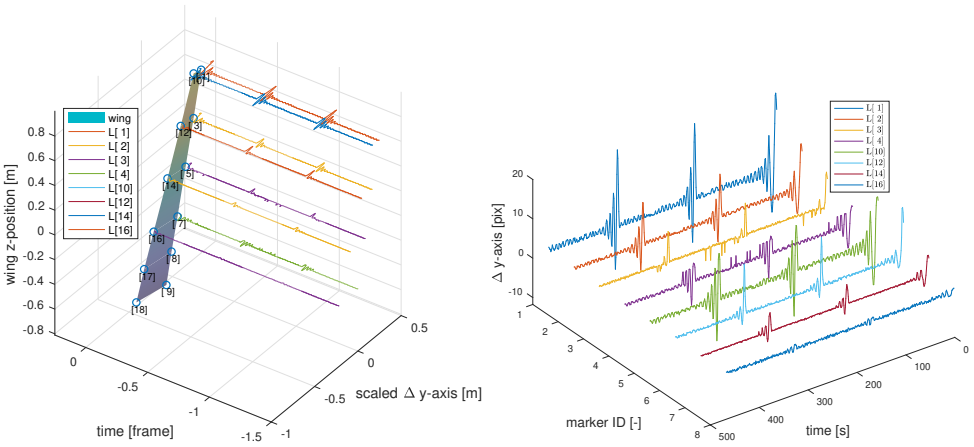
(d) Displacement in y of R1 sequence marker ID 7.

Figure I.1.19: Validation and comparison of DBSCAN and Disjoint-set data structure results for run R1, markers IDs 7 and 2.

I.1

I.1.4.3. Experimental Data Analysis with DBSCAN and Disjoint-Set Data Structure Clustering

The experimental data collected from the image sequences were processed with both tracking pipelines and compared to the laser vibrometer measured wing response, shown in Fig. I.1.11a. The tracking pipeline, depicted in Fig. I.1.24, was found to be capable of tracing the wing's motion by correctly clustering the markers. This was confirmed by the DBSCAN tracking result regarding pixel (x, y) locations in the images, shown in Fig. I.1.20. The plots show the time traces for the detected markers, and three occurrences of decaying sinusoidal responses can be observed from the output. The image y-displacement in pixels is arbitrarily scaled to show a comparative response to wing geometry. Exactly 3 gusts were introduced to the wing that produced a measurement during the image sequence, as shown in Fig. I.1.11 for a single gust.



(a) Wing 3D view and scaled image y-displacement.

(b) Image y-displacement relative to the baseline.

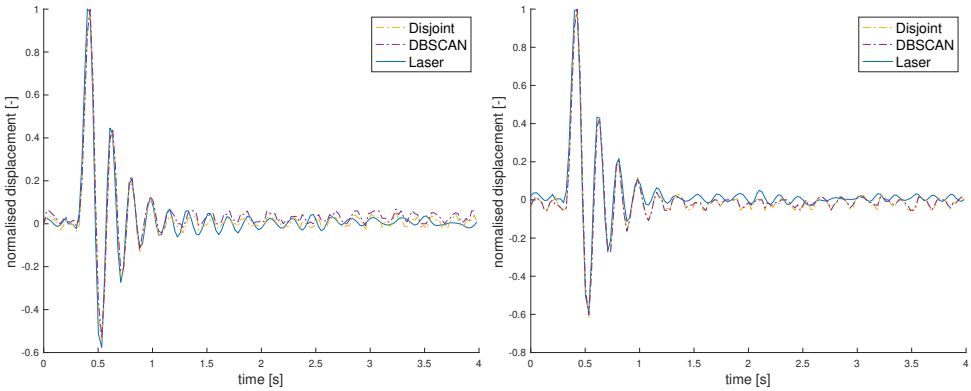
Figure I.1.20: Time series of marker y-displacement with DBSCAN across 469 image sequences of run R1.

Time-Domain Analysis

To further quantify the measured wing response and the tracked wing motion from the image sequence, comparisons were made in terms of displacements and frequency content. Fig. I.1.20 shows the comparison of the sampled wing displacement response extracted from the laser vibrometer (blue), the Disjoint-set data structure tracking pipeline (yellow) and the DBSCAN tracking pipeline (purple). In this figure, markers with high motion amplitude and relatively high (marker ID 7), as well as low (marker ID 8) validation errors, are compared. The response, shown here for both tracking pipelines, is the displacement in the y-direction, where the highest amplitudes were observed.

The laser vibrometer measured response (in meters) and the visually tracked motion (in pixels y direction) were normalised to allow effective comparison. The tracked responses in Fig. I.1.21 show good agreement with the laser measurements. Both tracking methods can capture the inherent damping of the wing with the correct amplitude decay and match the motion phase. The tracking results for marker ID 8 are slightly better compared to

marker ID 7, in agreement with the higher validation error measured for ID 7, as shown in Fig. I.1.18. However, as the error was mainly observed in the x-direction, differences in y are insignificant. DBSCAN shows better overall tracking performance. Furthermore, by observing the low amplitude oscillations after the initial gust onset, it can be seen that image tracking has a limit in terms of accuracy and resolution. In Fig. I.1.21a, it is observed that the tracking reduces in accuracy after the 4th peak (at 1 second) at roughly 10% of the maximum normalised amplitude, and the motion is overestimated by the tracking (from 2 seconds) by roughly 3% in the worst case. Considering that the maximum tip deflection is $\approx$ 33 mm (Fig. I.1.11a), the maximum motion tracking resolution is in the order of $\approx$ 0.1 mm. An obvious way to increase this resolution would be to use higher resolution cameras (>1.3 MP). However, this could directly increase the computational cost of the pipeline and reduce the maximum processing frame rate, with a subsequent penalty for the bandwidth of the image tracking.



(a) Comparison normalised response marker ID 8.

(b) Comparison normalised response marker ID 7.

Figure I.1.21: Comparison normalised response of the tip deflection for run R1 (laser versus image tracking).

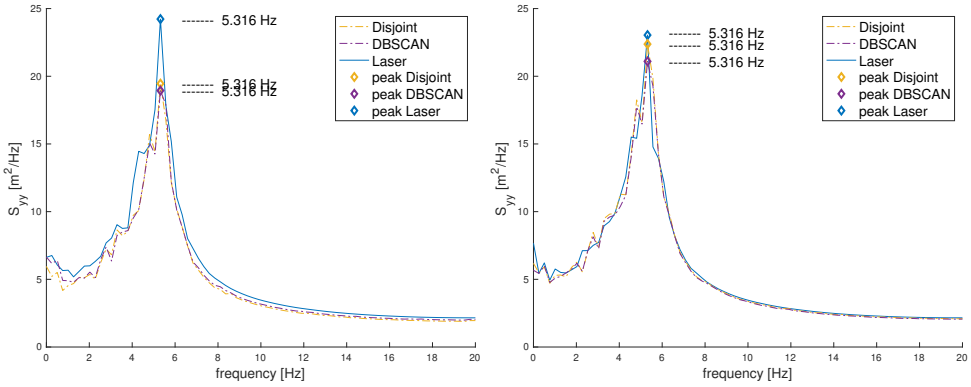
Frequency-Domain Analysis

Alternatively, a frequency domain analysis was performed, where the measurements were compared in terms of the measured output's power spectral densities (PSD). The main objective of this analysis was to understand whether the image tracking methods could correctly identify the frequency content of the measured signal compared to the reference measurement provided by the laser vibrometer. In this context, the power spectrum of the same response signal measured by three different methods provided sufficient grounds for the comparative assessment. This eliminated the need for a more elaborate frequency response analysis whereby the cross power spectrum of the output to input signals is computed as well in order to extract the system's frequency response function.

The auto-PSD of the output signal, S_{yy}, was calculated according to the following definition:

$$S_{yy}(\omega) = \int_{-\infty}^{\infty} R_{yy}(\tau) e^{-j\omega\tau} d\tau \tag{I.1.31}$$

where the integral in the expression is the Fourier transform of the auto-correlation function R_{yy} of the output signal (marker displacement).



(a) Auto-PSD of the response signal of marker ID 7.

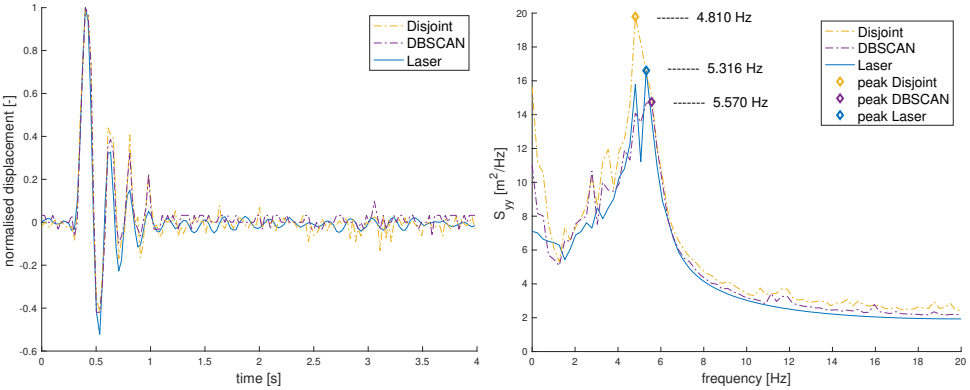
(b) Auto-PSD of the response signal of marker ID 8.

Figure I.1.22: Comparison of the frequency spectrum of the tip deflection response for run R1 (laser versus image tracking).

Figure I.1.22 shows the frequency content of the image sequence corresponding to the responses of marker IDs 8 and 7. As can be observed from Fig. I.1.22a), the image tracking methods are able to estimate the first resonant frequency of the wing, as for both markers, the peaks of the spectral densities align at 5.316 Hz. Furthermore, it is observed that, despite higher errors in the tracked response of ID 7, the resonance region is captured well in both tracking methods.

Here, the Disjoint-set data structure method estimates a slightly higher value for the power spectrum in the resonance region, which may better match the distribution obtained from the laser measurement. This can be explained by observing the response in Figs. I.1.21a and I.1.21b, showing that the oscillations tracked by the Disjoint-set data structure tend to estimate a higher power distribution around the resonance region compared to DBSCAN (i.e. higher sensitivity gain towards motion). For some markers, particularly those that exhibit lower motion activity (markers closer to the root), this can lead to overestimation of the response and — combined with a higher error — a shifted resonance peak. This is visible in Fig. I.1.23 for marker ID 3. Here, the Disjoint-set data structure overestimates the oscillations of the y-displacement (Fig. I.1.23a) and the resulting resonance peak is shifted from 5.316 to 4.810 Hz (Fig. I.1.23b). Although the RMSE of marker ID 3 is lower than ID 7 (Fig. I.1.18), the high RMSE of marker ID 7 is largely contributed by the x-displacement, hence for marker ID 3, a higher error in the y-displacement is probable. DBSCAN also has a slightly higher error for marker ID 3 but estimates the peak more accurately at 5.570 Hz.

Overall, the results are shown in Figures I.1.21 and I.1.22 suggest that the motion of an oscillating wing can be captured and analysed with relatively low-resolution cameras (1.3 megapixels). However, it is preferred to use markers exhibiting high motion amplitude (closer to the tip).



(a) Comparison normalised response marker ID 3.

(b) Auto-PSD of the response signal marker ID 3.

Figure I.1.23: Comparison of the response and the frequency spectrum of the marker ID 3, run R1 (laser versus image tracking).

Clustering Image Sequence

The clustering results of sequence R3, containing the marker loss due to LED failure, are shown in the lower row of Fig. I.1.24. Here, the dotted outline shows the initial contour at the baseline deflected shape before the gust hits the wing. The tracking pipeline schematic is shown below the figure. Despite the marker loss, DBSCAN is able to correctly deduce the number and the location of the markers without supervision in terms of the expected number of clusters.

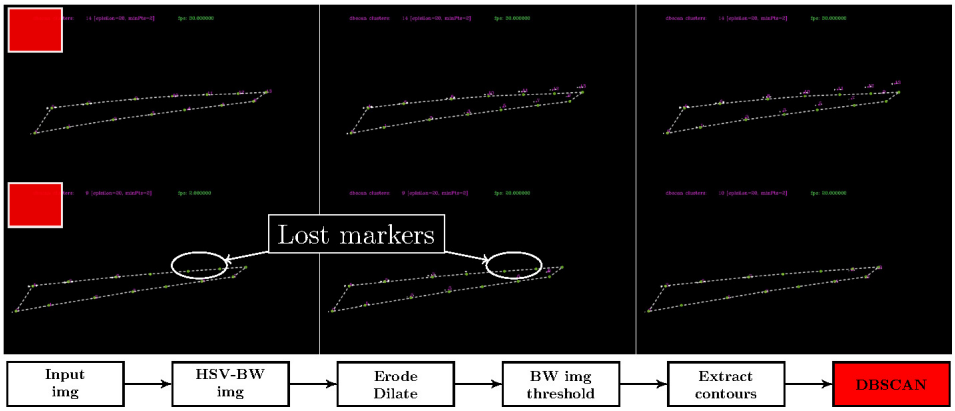


Figure I.1.24: Tracking sequence on input images from run R1 (upper row) and R3 (lower row).

I.1

I.1.4.4. Performance of DBSCAN^{-1} and the Limitation of DBSCAN Clustering

Assessment of DBSCAN Parameters

Figure I.1.25 shows the sensitivity of the tracking results to the $MaxPts$ parameter. When the parameter is set to 3, dictating that the Direct Density Reachability of the core points needs to contain a neighbourhood of at least three core points, markers 1 and 10 fail to meet these criteria and are no longer considered to be core points. The shapes (dataset D) in the extracted binary mask on which the clustering operation is done are influenced by the motion of the wing and the result of morphological filters (erode, dilate) performed after HSV filtering. As a result, at a given time instance, the 3-LED sub-units can be clotted together in one or two dots instead of three, never meeting the core point condition. In Fig. I.1.25, it is clearly illustrated that the cluster is found again once the units become more distinct; this is the case when the $MaxPts$ parameter is chosen to be 2, as shown in Fig. I.1.24.

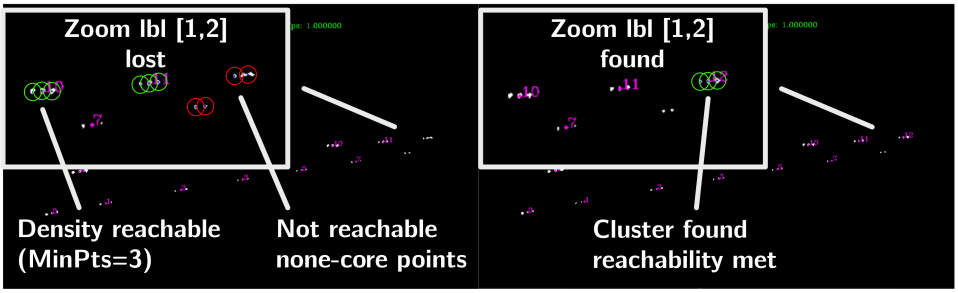


Figure I.1.25: Sensitivity of DBSCAN parameters. Snapshot of two frames from sequence R1, the frames are ≈ 0.025 seconds apart. The parameters are: $\epsilon = 20$, $MinPts = 3$.

Evaluation of Robustness Against Noise

The runs R1 and R3 were injected with Gaussian noise (mean of $\mu = 0$ and a standard deviation of $\sigma = 0.5$), and the tracking performance was evaluated. In Fig. I.1.26, a sequence is shown for tracking of frames 0, 50 and 100. The schematic of the corresponding tracking pipeline is provided at the bottom of the figure. The colour codes correspond to the operation steps performed in the pipeline throughout the sequence. From top to bottom, the rows represent input with noise image (grey), HSV filtering (green), threshold image (blue) and clustering result (red). The dotted outline shows the initial contour at the baseline deflected shape before the gust hits the wing.

As can be seen, the HSV filter combined with the morphological operations (erode and dilate) can cope well with the Gaussian noise. The morphological operations together with the HSV filter are, in fact, acting as a complex de-noising filter that produces a clean output which is in turn passed through as an input to DBSCAN. DBSCAN is then able to produce a robust result on the thresholded binary image, despite the high level of noise injected into the input image. For run R3, similar results were obtained, as presented in Sec. I.A1.2, Fig. I.A1.1.

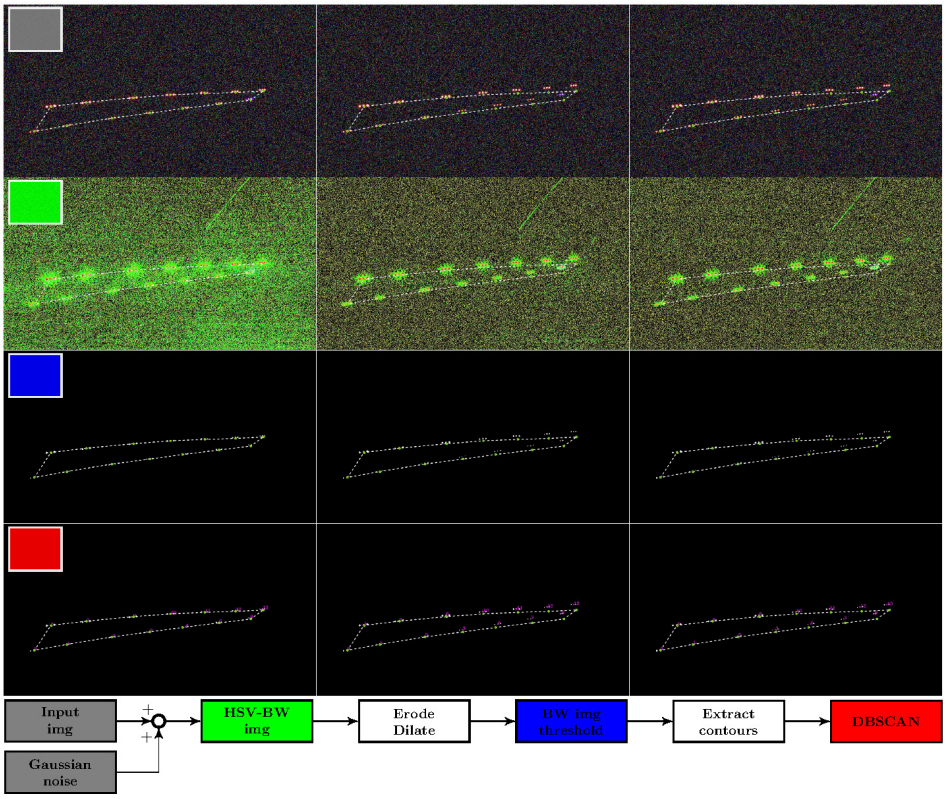


Figure I.1.26: Tracking sequence on input images (0,50,100) from run R1 with injected Gaussian noise ($\mu = 0$ and $\sigma = 0.5$).

Sensitivity of HSV Filtering and Morphological Operations

The benefit of the complex de-noising filter (HSV-morphological) with regard to noise and better clustering of input data was evident. There is, however, a condition where a combination of these filters can have an adverse effect. This is, in particular, the case in the presence of varying lighting conditions, such as the case for images with a light source, i.e. bright images, as per the definition in Sec. I.1.3.2.

In Fig. I.1.27, the effect of HSV filtering strategies is shown for images with a light source recorded by the trailing edge camera. The columns of images, from left to right, correspond to: (i) HSV filtering with morphological operations and no additional noise in the input (default case); (ii) same, but without morphological operations; (iii) same as condition (i), but with added noise in the input. The pipeline is shown on the right of Fig. I.1.27 corresponding to the rows' processing steps. As was observed in case (i), the default complex filter had a highly *adverse effect* and the marker contours required for clustering were largely *erased*. This happens as the HSV filter produces high density scattered noise particles in the output image (the colour content in the lighter image is higher),

and the subsequent morphological erode filter, due to the local min-max operation, ends up removing the relevant data from the output image alongside the noise. This becomes clear when compared to case (ii), where no morphological operations were applied: the contours of the markers are retained, albeit with much higher noise content compared to the nominal R1 dataset (dark images). This results in the false detection of markers (centre area of the wing) in the clustering step. When additional noise is added (without morphological operations) in case (iii), the result is similar but noisier, as expected. This analysis indicates that morphological operations are not always possible or beneficial and highlights the relevance and need for a novel clustering approach with DBSCAN^{-1}.

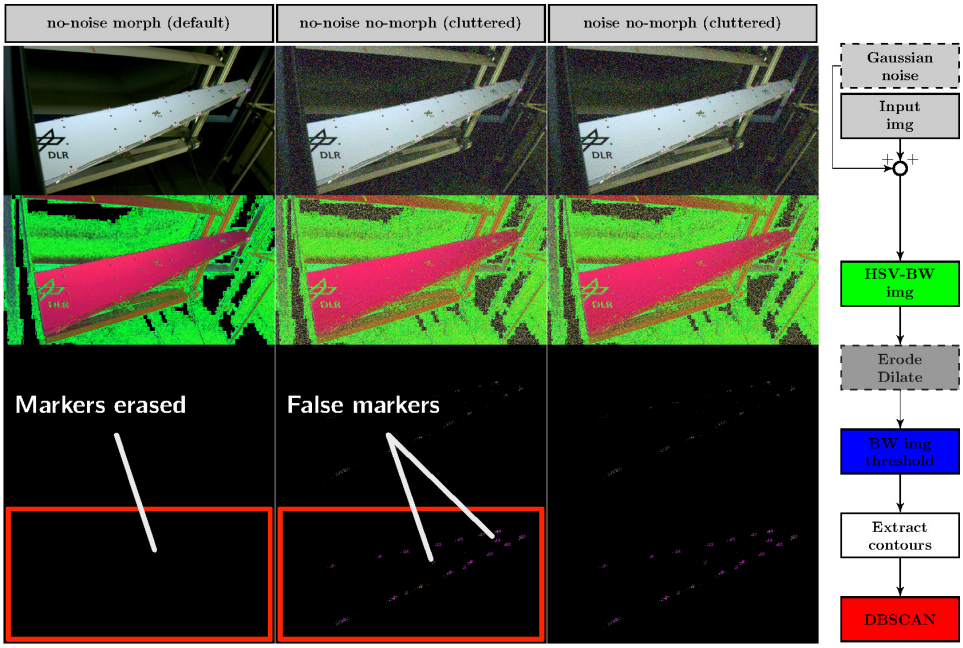


Figure I.1.27: Sensitivity of HSV filtering and morphological operations to varying lighting conditions.

DBSCAN^{-1} in the Presence of Noise without Morphological Operations
This analysis investigated the robustness of DBSCAN without additional de-noising filters. The same experiment was run to see how well DBSCAN would fare when exposed to more noise and fewer filtering steps, particularly when morphological operations were removed. These operations proved highly capable of filtering out the remnants of Gaussian noise after HSV operation, marked as the blue filtering block in Fig. I.1.26. It was, therefore, interesting to examine the effect of removing the morphological filter block.

The nominal DBSCAN result using the same parameters ($\epsilon = 20$ pix, $MinPts = 2$) is shown in Fig. I.1.28. The schematic of the clustering pipeline at the bottom shows that the most critical change is in disabling the morphological operations (erode and dilate, in transparent grey). As a result, the thresholded binary image contains noisy speckles

detected as core points in DBSCAN. In the third image (red colour code), the clustering fails to detect the markers properly. Instead, a large number of clusters is detected. This is clearly illustrated in the rightmost image, where each point is identified with a numeric label belonging to the cluster-ID, where magenta labels represent valid clusters and grey (-1) labels — outliers or noise. This can be partially remedied by further tuning of the DBSCAN parameters. However, the additional noise in the threshold image will continue to produce problems for correctly detecting the remainder of the markers.

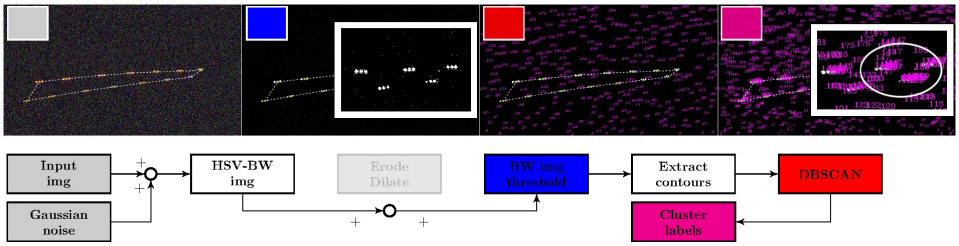


Figure I.1.28: Tracking sequence on input images (0,50,100) from run R1 with injected Gaussian noise ($\mu = 0$ and $\sigma = 0.5$) and disabled morphological operations (transparent block).

A better approach would be to use DBSCAN clustering differently. In the Sec. I.1.2.4, a methodology was proposed to approach the DBSCAN clustering from a novel viewpoint. DBSCAN is known for its ability to discard points that are not part of a cluster as noise. Instead of looking for cluster centres, it was proposed to use DBSCAN in an inverse fashion to detect *noise* (non-core points). In this approach, $MinPts$ is set to 1, allowing to maximise the number of points forming a cluster and, instead, to capture the desired clusters by tuning the ϵ and $MaxPts$ parameters.

The result of this analysis, with $\epsilon = 20$, $MinPts = 1$ and $MaxPts = 8$, is shown in Fig. I.1.29. The input to DBSCAN is the same as in the sequence of Fig. I.1.28, except, as schematised at the bottom of the figure, the inverse DBSCAN filter (cyan block) is applied instead. As a result, the desired clusters (markers) are identified as noise (obtaining a grey -1 label), and the rest of the points are identified as valid clusters. At the same time, the nominal DBSCAN (Fig. I.1.28) was unable to deal with this without the additional de-noising filter. Thus, DBSCAN^{-1} has an advantage over the nominal DBSCAN in this particular scenario.

In essence, the DBSCAN^{-1} approach actively looks for noise, discarding the actual clusters. Subsequently, the clusters can be retrieved by an additional step where the nominal DBSCAN is applied again. To this end, a new parameter, $MaxPts$, was introduced, putting a cap on the number of reachable core points within a cluster. Since noise will be randomly and densely clustered together, the probability is high (for most noise models) that noise particles will be surrounded by a dense number of other noise particles within an arbitrary ϵ neighbourhood. It is important to note that this condition holds for $MinPts = 1$, such that the number of points forming a cluster is maximised.

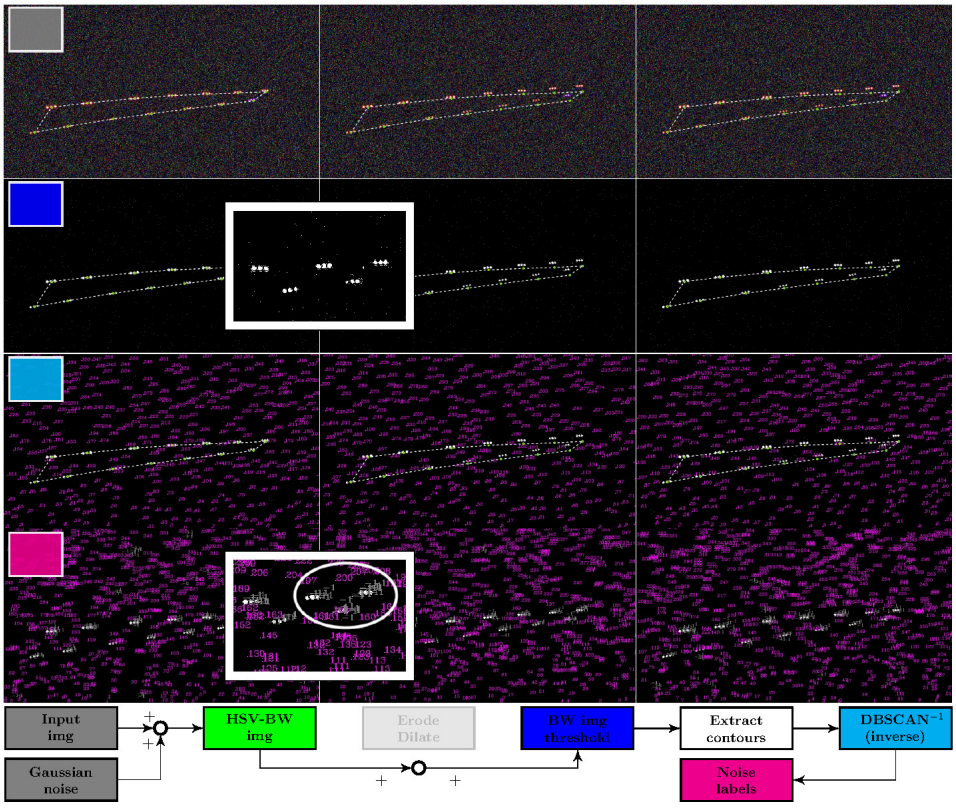


Figure I.1.29: Tracking sequence on input images (0,50,100) from run R1 with injected Gaussian noise ($\mu = 0$ and $\sigma = 0.5$), and DBSCAN^{-1}.

I.1.4.5. Effect of Image Thresholding

Variations of light and motion activity of the object make the task of obtaining a good thresholding challenging. This study investigated several thresholding approaches, namely, global normalisation, baseline normalisation and adaptive global thresholding using Otsu's method. The analysis is shown in Fig. I.1.30. Here, the input images from run R1 were converted to greyscale and the thresholding strategy was applied as described in Sec. I.1.2.2. The corresponding pixel intensity map obtained after thresholding is shown in the view of the 3D (left column) and top (right column). The regions indicated with dark peaks correspond to high occurrence pixel regions passing the threshold and thus high pixel activity. The scatter points in red correspond to the marker centres' travel across the image sequence and high wing motion activity. The collected range of input images was arbitrarily chosen at intervals as a continuous vector:

$$N_n = \begin{bmatrix} 1 & 4 & 11 & 90 & 95 & 160 & 168 & 274 & 326 & 330 & 424 & 469 \end{bmatrix}$$

Global Normalisation Thresholding

The results of global normalisation are shown in Fig. I.1.30a. Here, the full image sequence was converted to greyscale and normalised in the $0 - 255$ range. Then, a threshold of $\tau_{th} = 9 \cdot 10^{-5}$ was applied. Subsequently, the pixel intensities across time were summed over the input vector N_n:

$$\sum_{n=0}^{N} G(x,y)_n = \sum_{n=0}^{N} f_{norm}(I(x,y)_n) \tag{I.1.32}$$

A 3D view of the accumulated pixel intensity sum is shown in Fig. I.1.30a. It can be observed that the so-called *footprint* of the markers, indicated by the red line, corresponds to high pixel activity in the spatial domain across the input sequence. This footprint corresponds to the full sequence from Fig. I.1.20a projected to the spatial domain of the image. The pixel activity is indicated by high-intensity values retained after thresholding (dark peaks). For a continuous input sequence, correct thresholding should characterise the moving foreground as high intensity and filter out the static background. This appears to be the case here; nonetheless, a disbalance in the height of the peaks was observed. The high peaks are observed at the location of the bottom markers (below ID's 14-5 in Fig. I.1.12a), indicating that high-intensity values were captured repeatedly at these locations. Observing the top view plot in Fig. I.1.30b, it can be deduced that the highest motion amplitudes belong to the wingtip, meaning that the outline around the bottom markers should have been more cleanly filtered by the threshold due to the low motion activity.

Baseline Thresholding

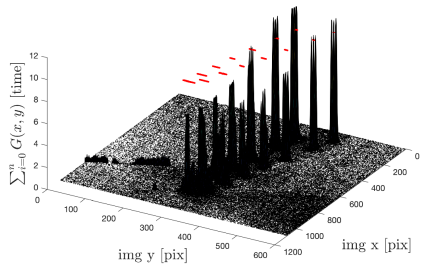
The results of the baseline thresholding are shown in Figs. I.1.30c and I.1.30d. The main observation of baseline thresholding is that the balance between the peak heights is retained, and the static areas around the bottom markers are correctly filtered. However, since the differences in pixel intensities are now shifted closer together, the background is noisier and distinct static patterns are picked up due to a higher sensitivity to the threshold parameter, which is undesirable. This can be partially remedied by adjusting the threshold parameter.

Adaptive Otsu's Thresholding

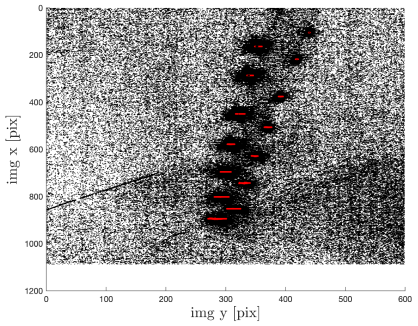
The results of Otsu's thresholding are shown in Figs. I.1.30e and I.1.30f, where a clear definition of a (moving) foreground can be observed. The high pixel intensities are correctly assigned to the marker centres alone, and the background is entirely absent. Figure I.1.30d shows an obvious agreement between the marker footprint and pixel intensity. Also, the height of the peaks is more balanced, meaning that the static areas are efficiently removed. The results indicated that Otsu's method is the cleanest approach for thresholding for this given dataset.

I.1.4.6. Adaptive Thresholding with Time-Spatial Frequency Data

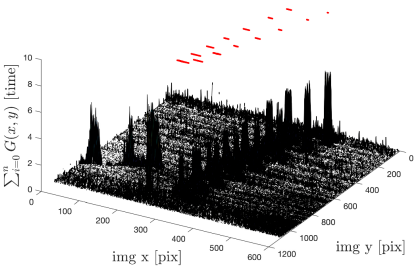
The methodology explained previously along with the results on the assessment of image thresholding (Figs. I.1.30a, I.1.30c and I.1.30e), as well as the time traces (Fig. I.1.20a), are



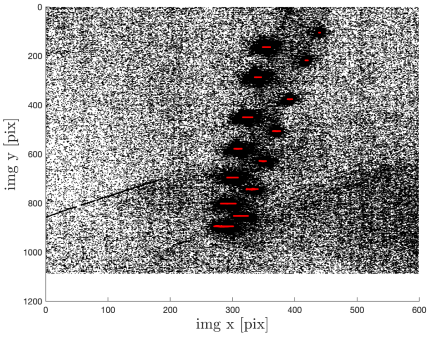
(a) 3D view global normalisation thresholding.



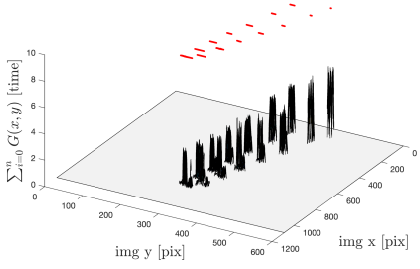
(b) Top view global normalisation thresholding.



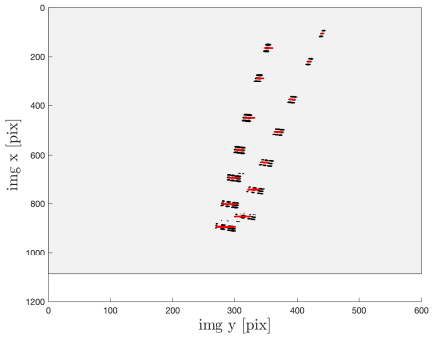
(c) 3D view view baseline thresholding.



(d) Top view baseline thresholding.



(e) 3D view Otsu thresholding.



(f) Top view Otsu thresholding.

Figure I.1.30: 3D (right column) and top (left column) view of the accumulated pixel intensity values after thresholding across the 469 frame sequences of R1.

indicative of an interconnection between segmentation and clustering. However, the gap is in connecting these processes in time. That is, to find a relationship between segmentation and clustering parameters to adaptively obtain an optimal marker label detection through time for a sequence of images.

To this end, a method implementing the Sliding Discrete Fourier Transform (SDFT) [50] is suggested. This approach is illustrated in Fig. I.1.31. Here an image sequence is shown with markers indicated in red (not in true scale). A point P_1 corresponding to a marker label with coordinates x_i and y_i, represents a displacement signal in x, y pixel values in time domain. Consequently, its movement corresponds to a *footprint* in the spatial domain. The grey threshold value is adjusted to capture P_1 as it moves in the footprint. As more data reflecting the motion of P_1 is collected gradually in the SDFT time window, the peak amplitude becomes more prominent at a distinct resonance frequency value ω_n. This continuously updated knowledge of the wing dynamics projected on the spatial image domain in terms of high pixel activity regions can consequently be used to adjust the parameters of the marker detection pipeline.

The nature of the SDFT allows the method to be used in real-time at a low cost. Implementing this approach would effectively enable the use of the pipeline as a robust tracking algorithm. In a follow-up study, the role and the benefit of the SDFT are to be further investigated.

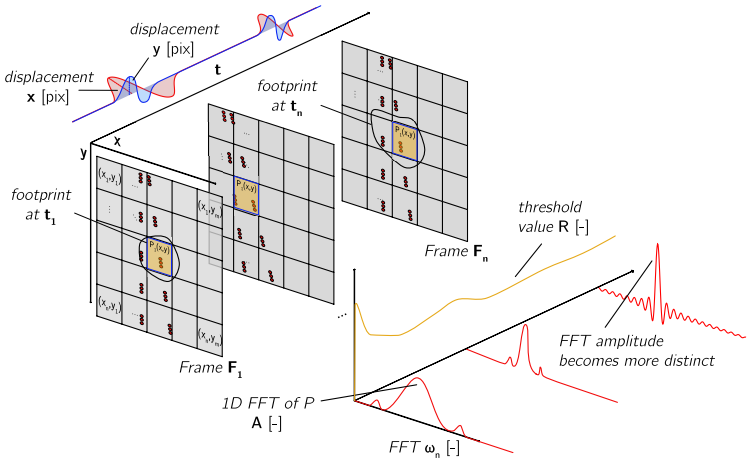


Figure I.1.31: Simplified spatial and time representation of the SDFT and thresholding adaptation for a sequence of images.

I.1.5. Conclusions and Recommendations

In this study, an image tracking pipeline was developed using a robust machine learning approach, with the aim to (i) automatically label visual markers and (ii) investigate a non-invasive state estimation approach for online control applications of flexible aircraft wings.

The pipeline consisted of an image segmentation, where a mask for clustering operations was obtained with an HSV colour filter and a threshold filter using morphological

operations (erode and dilate). Subsequently, the mask was clustered using unsupervised clustering with DBSCAN [21]. DBSCAN was compared to another unsupervised clustering method, the Disjoint-set data structure [22] by processing (i) performance data obtained from a performance test with randomly generated cluster data and (ii) experimental data obtained from the measurements of the wing response undergoing oscillatory motion under gust excitations in the OJF (Open Jet Facility) wind tunnel at the Delft University of Technology.

The DBSCAN pipeline was found to be more reliable and robust in terms of accuracy and resilience against the nose in the input image, which was confirmed both with the performance data and the experimental data. Next to showing an overall better tracking capability compared to the Disjoint-set data structure, an error of ≈ 1 pixel was observed for most of the markers clustered with DBSCAN concerning the validation dataset.

An essential shortcoming of the de-noising HSV-morphological segmentation filter was highlighted with regard to sensitivity to noise and variations in image illumination. More specifically, that (i) the clustering performance degraded without morphological operations, and (ii) the mask for the clustering operations could be erased by morphological operations under certain lighting conditions (high illumination). This suggested that morphological operations are not always possible or beneficial and highlighted the relevance and the need for a novel clustering approach.

To tackle this problem, a novel formulation of DBSCAN was proposed, namely, the inverse DBSCAN (denoted DBSCAN^{-1}), where the clustering problem is reformulated into a noise filtering problem. Instead of rejecting, this approach explicitly detects the noise, making the clustering an implicit task. The experimental dataset was processed using the DBSCAN^{-1} pipeline, and it was shown that the actual clusters were successfully identified and isolated from the noise in the image. After isolating the clusters, DBSCAN^{-1} must be followed by an additional nominal DBSCAN clustering to extract the exact location of the markers. The final nominal DBSCAN can be done at a significantly lower computational cost due to the removed noise. Further studies are required to assess the performance gain of DBSCAN^{-1} compared to additional filtering steps in various lighting conditions.

In conclusion, the time- and frequency-domain analyses of the experimental data suggested that the motion of an oscillating wing can be adequately captured with relatively low-resolution cameras (1.3 megapixels) in the proposed tracking pipeline. It was suggested that the SDFT could also extract the frequency content of the image sequence, and the image thresholding step was continuously adapted to produce a better tracking result. The accuracy and motion resolution could be further improved by increasing the camera's resolution and utilising markers less sensitive to lighting conditions, such as active Infrared (IR) markers. However, a thorough trade-off is essential, as this would generally increase the computational cost of the tracking pipeline and reduce the maximum processing frame rate, subsequently reducing the tracking bandwidth.

References

[1] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive Real-Time Clustering Method for Dynamic Visual Tracking of Very Flexible Wings," *Journal of Aerospace Information Systems*, vol. 18, pp. 58–79, jan 2021.

[2] A. W. Burner and T. Liu, "Videogrammetric model deformation measurement technique," *Journal of Aircraft*, vol. 38, no. 4, pp. 745–754, 2001.

[3] P. I. Corke, "Visual Control of Robot Manipulators – a Review," in *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, pp. 1–31, World Scientific, 1993.

[4] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern recognition letters*, vol. 34, no. 1, pp. 3–19, 2013.

[5] A. N. Belbachir, *Smart cameras*, vol. 2. Springer, 2010.

[6] Y. Choi, M. Martel, S. Briceno, and D. Mavris, "Multi-UAV trajectory optimization and deep learning-based imagery analysis for a UAS-based inventory tracking solution," in *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics (AIAA), 2019.

[7] Y. Hu, Y. Cao, M. Ding, and L. Zhuang, "Airport detection for fixed-wing unmanned aerial vehicle landing using a hierarchical architecture," *Journal of Aerospace Information Systems*, vol. 16, pp. 214–223, jun 2019.

[8] K. Abu-Jbara, G. Sundaramorthi, and C. Claudel, "Fusing vision and inertial sensors for robust runway detection and tracking," *Journal of Guidance, Control, and Dynamics*, vol. 41, pp. 1929–1946, sep 2018.

[9] P. Agrawa, A. Ratnoo, and D. Ghose, "Image segmentation-based unmanned aerial vehicle safe navigation," *Journal of Aerospace Information Systems*, vol. 14, pp. 391–410, jul 2017.

[10] J. Valasek, D. Famularo, and M. Marwaha, "Fault-Tolerant adaptive model inversion control for vision-based autonomous air refueling," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 6, pp. 1336–1347, 2017.

[11] C. Parsons, Z. Paulson, S. Nykl, W. Dallman, B. G. Woolley, and J. Pecarina, "Analysis of simulated imagery for real-time vision-based automated aerial refueling," *Journal of Aerospace Information Systems*, vol. 16, pp. 77–93, mar 2019.

[12] R. D. Abousleiman, O. A. Rawashdeh, and M. R. Siadat, "Statistical algorithm for attitude estimation from real-time aerial video," *Journal of Aerospace Computing, Information and Communication*, vol. 7, no. 10, pp. 322–337, 2010.

[13] T. A. Weisshaar, "Morphing aircraft systems: Historical perspectives and future challenges," *Journal of Aircraft*, vol. 50, no. 2, pp. 337–353, 2013.

[14] L. Cai, L. He, Y. Xu, Y. Zhao, and X. Yang, "Multi-object detection and tracking by stereo vision," *Pattern Recognition*, vol. 43, pp. 4028–4041, dec 2010.

[15] M. M. Al-Isawi and J. Z. Sasiadek, "Control of flexible wing UAV using stereo camera," in *GeoPlanet: Earth and Planetary Sciences*, pp. 97–120, Springer Verlag, 2019.

I.1

[16] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Parallel Real-Time Tracking and 3D Reconstruction with TBB for Intelligent Control and Smart Sensing Framework," in *AIAA Scitech 2020 Forum*, p. 2252, American Institute of Aeronautics and Astronautics (AIAA), jan 2020.

[17] B. Borah and D. K. Bhattacharyya, "An Improved Sampling-Based DBSCAN for Large Spatial Databases," in *Proceedings of International Conference on Intelligent Sensing and Information Processing, ICISIP 2004*, pp. 92–96, 2004.

[18] A. Smiti and Z. Elouedi, "DBSCAN-GM: An improved clustering method based on Gaussian Means and DBSCAN techniques," in *INES 2012 - IEEE 16th International Conference on Intelligent Engineering Systems, Proceedings*, pp. 573–578, 2012.

[19] M. Noticewala and D. Vaghela, "MR-IDBSCAN: Efficient Parallel Incremental DBSCAN algorithm using MapReduce," Tech. Rep. 4, Parul Institute of Technology, 2014.

[20] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," 2007.

[21] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, vol. 96, pp. 226–231, 1996.

[22] B. A. Galler and M. J. Fisher, "An improved equivalence algorithm," *Communications of the ACM*, vol. 7, pp. 301–303, may 1964.

[23] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval," in *IEEE International Conference on Image Processing*, vol. 2, pp. II—-II, 2002.

[24] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, Jan. 1979.

[25] P. Lancelot, J. Sodja, and R. De Breuker, "Investigation of the unsteady flow over a wing under gust excitation," *17th International Forum on Aeroelasticity and Structural Dynamics, IFASD 2017*, vol. 2017-June, 2017.

[26] A. Bovik, *Handbook of Image and Video Processing*. Academic press, 2005.

[27] M. Cai, J. Song, and M. R. Lyu, "A new approach for video text detection," in *IEEE International Conference on Image Processing*, vol. 1, pp. I–117–I–120, Ieee, 2002.

[28] S. Suzuki and K. A. Be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics and Image Processing*, vol. 30, pp. 32–46, apr 1985.

[29] R. Acharya, *Multidimensional image analysis and mathematical morphology*. Academic Press, Inc., 1989.

[30] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[31] J. Liu, W. Li, and Y. Tian, "Automatic thresholding of gray-level pictures using two-dimension Otsu method," in *1991 International Conference on Circuits and Systems*, pp. 325–327, Institute of Electrical and Electronics Engineers (IEEE), dec 2002.

[32] M. Vala and A. Baxi, "A review on Otsu image segmentation algorithm," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 2, pp. pp: 387–389, 2013.

[33] J. Kittler and J. Illingworth, "On Threshold Selection Using Clustering Criteria," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-15, no. 5, pp. 652–655, 1985.

[34] G. H. Shah, "An improved DBSCAN, a density based clustering algorithm with parameter selection for high dimensional data sets," in *3rd Nirma University International Conference on Engineering, NUiCONE 2012*, pp. 1–6, 2012.

[35] R. A. Jarvis, "On the identification of the convex hull of a finite set of points in the plane," *Information Processing Letters*, vol. 2, pp. 18–21, mar 1973.

[36] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive Real-Time Clustering Method for Dynamic Visual Tracking of Very Flexible Wings," in *AIAA Scitech 2020 Forum*, p. 2250, American Institute of Aeronautics and Astronautics (AIAA), jan 2020.

[37] K. Jongkind, A. Falkmann, and H. van der Veer, "Open Jet Facility," 2020.

[38] Polytec, "Polytec Single-Point Vibrometers," 2020.

[39] M. Ritter, Y. M. Meddaikar, and J. K. Dillinger, "Static and dynamic aeroelastic validation of a flexible forward swept composite wing," in *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2017*, p. 0637, 2017.

[40] Basler AG, "Basler ace acA1300-30gm - Area Scan Camera," 2019.

[41] Computar, "Computar Machine Vision Lens Catalog," tech. rep., Computar, 2017.

[42] "Embedded Computer Vision Real Time? Nvidia Jetson TX2 + VisionWorks toolkit - Myzhar's MyzharBot and more...," 2020.

[43] G. Bradski, "The OpenCV Library," *Dr Dobbs Journal of Software Tools*, vol. 25, pp. 120–125, 2000.

[44] NVIDIA, "Jetson TX2 GStreamer guide - Google Search," 2020.

[45] GStreamer, "GStreamer: open source multimedia framework," 2020.

[46] T. Mkhoyan, "tmkhoyan/cvyamlParser: Initial public release," nov 2019.

[47] T. Mkhoyan, "tmkhoyan/adaptiveClusteringTracker: Initial public release," 2019.

[48] T. Mkhoyan, "tmkhoyan/autoLabelTool: Initial public release," may 2020.

I.1

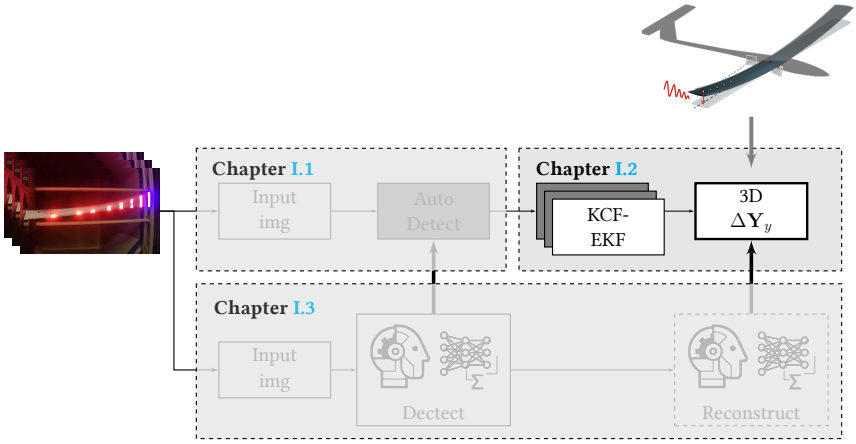
[49] T. Mkhoyan, "tmkhoyan/adaptivePerformanceTest: initial release," May 2020.

[50] E. Jacobsen and R. Lyons, "The sliding DFT," *IEEE Signal Processing Magazine*, vol. 20, pp. 74–80, mar 2003.

I.2

State Estimation and Real-Time Tracking

In this Chapter, a method is proposed for robust visual tracking and reconstruction of flexible wing motion for controller feedback. The method consists of a tracker pair composed of a purely visual filter, a high-speed Kernelised Correlation Filter (KCF), paired with an Augmented Extended Kalman Filter (AEKF), allowing to adaptively estimate the states and parameters of the system exhibiting oscillatory motion. The method consisting of KCF-EKF, without Kalman Filter augmentation, was validated experimentally on a real-time image stream of a very flexible wing subjected to gust excitation in the OJF (Open Jet Facility) at the Delft University of Technology [1].



This Chapter is based on the following peer-reviewed conference paper:
T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive State Estimation and Real-time Tracking of Aeroelastic Wings With Augmented Kalman Filter and Kernelized Correlation Filter," in *AIAA Scitech 2021 Forum*, p. 0666, 2021

I.2

This Chapter investigates a non-invasive vision-based approach for motion and shape reconstruction of a flexible wing for implementation in a control-feedback loop for gust and manoeuvre load alleviation objectives. To achieve this, an approach is proposed consisting of a visual tracking method, with fuselage-mounted cameras observing the wing's motion in real-time. High-speed visual tracking with correlation filters such as KCF (Kernelised Correlation Filter) facilitates efficiently and robustly correlation between two samples with Kernelised linear regression. A purely visual tracking filter, however, does not contain information regarding the system's dynamics subject to tracking and may fail under marker loss and occlusion. To increase the robustness of the tracking, an EKF (Extended Kalman Filter) is added to the tracking filter, acting as a KCF-EKF tracking couple. The Kalman Filter is further augmented into augmented Kalman Filter form to allow joint online estimation of the model states and parameters. This proposed tracking approach is used to adaptively reconstruct in real-time the motion of a very flexible wing subject to gust excitation in the OJF (Open Jet Facility) wind tunnel at the Delft University of Technology. The method shows a good agreement with the time and frequency domain analysis of the reference data measured by a laser vibrometer and demonstrates the effectiveness of the KCF-AEKF couple under marker loss and model uncertainties for a model-free control approach.

I.2.1. Introduction

The tendency towards increasingly flexible aircraft opens the possibility for better structural and aerodynamic efficiency. However, with increasing flexibility and lighter weight, the susceptibility to higher dynamic loads is increased. Aeroservoelastic control design is one of the fields where the flexibility of the structure and its interaction with aerodynamic loads is closely coupled and challenging. In such controller designs, monitoring the entire wingspan can be crucial for adequately delegating control actions. This may involve installing many conventional accelerometers that are likely subject to noise and bias, suffer from lack of space, must deal with certification requirements and face challenges associated with correct geometric placement. An intelligent sensing approach is desired for those examples of wing structures that rely on novel types of sensors to provide feedback to an intelligent controller.

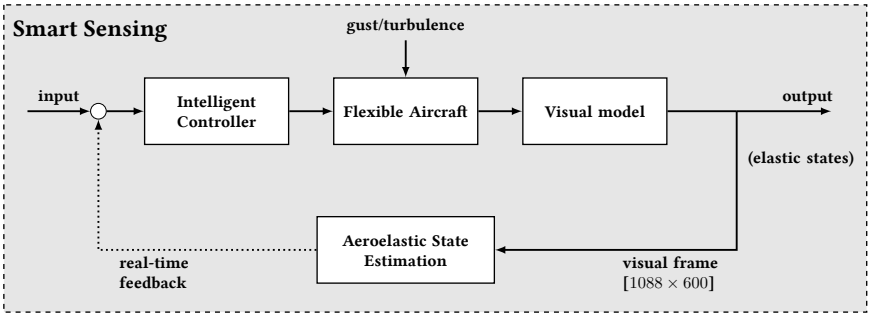


Figure I.2.1: State estimation setup with visual tracking in a controller scheme.

A solution that can significantly reduce the complexity associated with hardware installation, and provide the flexibility needed for employing novel state estimation methods, is aeroelastic state estimation by visual methods. An illustration of aeroelastic state estimation using visual data is shown in Fig. I.2.1.

The use of visual information for observing deformations has been successfully implemented on wind tunnel models in early studies [2] and has also seen wide application in robot manipulation [3]. However, in recent years, the capability of on-board computation and the camera quality has immensely increased, while the hardware has become more compact [4, 5]. This opens the door for numerous embedded applications using a camera as a sensor. In particular, fuselage-mounted camera systems can provide significant advantages for flexible aircraft systems, save costs associated with installation and certification, and have the potential of being non-invasive and universally applicable. This type of smart sensing system is also essential for over-actuated, and over-sensed morphing designs such as the SmartX [6]. Furthermore, image data is a rich source of information; collected over a period of time, it unlocks the opportunity to approach the state estimation from a new perspective using machine learning methods.

While many suitable tracking methods exist for marker detection, purely visual tracking methods have no knowledge of the underlying dynamics of the system. The disadvantage is that a visual filter alone may fail in the presence of occlusions in the image, show sensitivity to lighting conditions and produce high errors under large deformations of the structure. To correctly estimate the motion of the structure, an adaptive approach is required that is aware of the system exhibiting dynamic behaviour and attempts to estimate system parameters.

To account for this shortcoming, this Chapter describes a tracking method consisting of a purely visual filter, a high-speed Kernelised Correlation Filter (KCF), with an Augmented Extended Kalman Filter (AEKF). The combined tracking KCF-AEKF filter adaptive estimates online both the state and the system's parameters subject to disturbance, exhibiting oscillatory motion. Adding the AEKF and parameter estimation reduces the model's dependency, making this approach very suitable for closed-loop control of uncertain systems and very flexible systems exhibiting nonlinear responses. The method is validated experimentally on a real-time image stream of a very flexible wing subjected to gust excitation in the OJF (Open Jet Facility) at the Delft University of Technology.

The main contributions of this Chapter are threefold. First, it formulates a visual tracking approach coupled with a high-speed Kernelised Correlation Filter and Extended and Augmented Kalman Filters to increase the robustness of the tracking filter. Second, gust parameter sensitivity and wing response analysis are conducted in the context of wing shape and motion reconstruction, such that the tracking performance can be improved. Finally, it presents the development and realisation of a non-invasive smart sensing approach for shape and motion reconstruction of flexible and morphing wings, using an efficient algorithm, processing and embedded hardware implementation.

This Chapter is structured as follows. First, the methodology is presented in Sec. I.2.2. More specifically, the visual-based Kernelised Correlation Filter is presented in Sec. I.2.2.1. Next, Sec. I.2.2.2 investigates how Extended and Augmented Kalman Filters can be coupled to the visual filter for state and parameter estimation. Section I.2.2.3 describes the complete image tracking pipeline used for wing shape and motion reconstruction. This is followed by Sec. I.2.3 describing the apparatus, the wing model and the experimental system constructed for tracking visible LED markers. Motion conditions and experimental procedure are discussed in Sec. I.2.3.3. Sec. I.2.4 discusses the main results conducted in the OJF wind tunnel. Finally, Sec. I.2.5 closes off with conclusions and recommendations.

I.2.2. Methodology

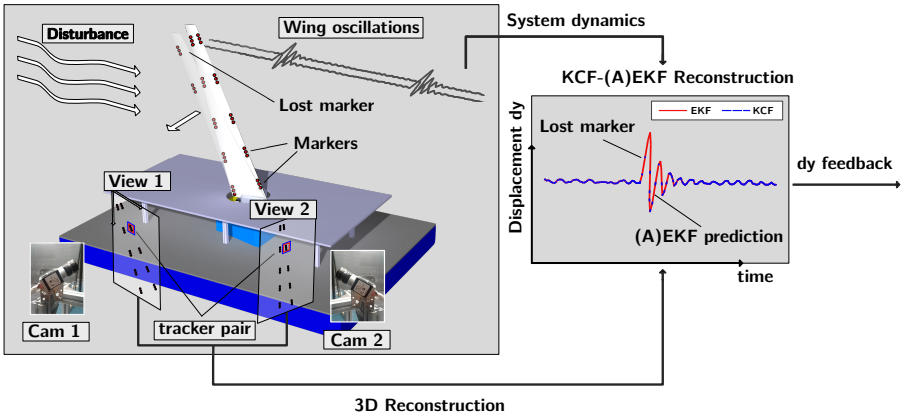


Figure I.2.2: Illustration of the wing tracking and reconstruction approach using KCF-(A)EKF.

The method comprises a visual tracking method combined with adaptive state estimation. The visual tracking uses state-of-the-art high-speed KCF (Kernelised Correlation Filter). The tracker is augmented with AEKF (Augmented Extended Kalman Filter), allowing online estimation of augmented time-varying mass, stiffness, and damping states. Furthermore, the visual system provides the state and state derivative feedback signals in real-time.

I.2.2.1. KCF: Visual Filtering

KCF belongs to a family of correlation filters that aims to find the correlation between two samples. When applied to a tracking problem, the correlation filter observes the correlation of the original target object defined in a patch moving through future sample images. Regular correlation filters do not allow updating of the initial target object. KCF, in contrast, updates the model of the object online using a Kernelised linear regression, thereby retaining robustness against object mutation (change of appearance through samples). Another property of the KCF is the utilisation of the properties of circulant matrices and kernel functions, reducing the computation to an order $O(log(n))$ versus $O(n^3)$ for typical correlation filters. This allows the KCF tracker to reach high speeds and reduce the memory storage for the computation. Below is a summary of the KCF methodology [7].

Summary of the KCF Methodology

The aim of the tracking problem is to learn to predict the mutation, presence or absence of the target object in the future frames. The learning method in KCF is based on the linear ridge regression [7]:

$$\min_{w} \sum_{i=1}^{n} (f(\boldsymbol{x}_i) - y_i)^2 + \lambda ||\boldsymbol{w}||^2 \tag{I.2.1}$$

Where x_i are the samples, y_i are the regression targets, w is a vector of weights, λ is the regularisation parameter to prevent overfitting, and the objective is to find a function $f((z))$ that minimises the squared error. The latter function typically adapts the following model:

$$f(\boldsymbol{z}) = (w)^T \boldsymbol{z} \tag{I.2.2}$$

The minimisation can be formulated in the closed-form, which can be represented as a convex function with a unique solution as:

$$\boldsymbol{w} = (X^T X + \lambda I)^{-1} X^T \boldsymbol{y} \tag{I.2.3}$$

Here the matrix X contains the vector of sample images x_i per row, $\boldsymbol{y}$ contains the regression targets y_i as elements, and I is the identity matrix. The above formulation involves solving a large system of linear equations, which can work for a problem for real-time tracking. To elevate this, KCF uses circulant matrices composed of shifted samples of $\boldsymbol{x}$ of the object of interest and efficiently computes all possible cyclic shifts in the Fourier domain. To transition into the Fourier domain, the term X^T is replaced by the Hermitian transpose $X^H = (X*)^T$ and $X*$ represents the complex conjugate of X. The pattern by which the circulant matrices are generated is deterministic and fully determined by the generating vector x. A powerful property of the circulant matrices arising from their inherent structure is that they become diagonal in the Fourier domain. Denoting the Discrete Fourier Transform (DFT) as $\hat{\boldsymbol{x}} = \mathcal{F}(\boldsymbol{x})$, and matrix F as the constant DFT matrix not depending on input vector, the resulting matrix X in the Fourier domain becomes [8]:

$$X = F diag(\hat{\boldsymbol{x}}) F^H \tag{I.2.4}$$

This formulation allows the computational convenience of element-wise operation as the matrices are diagonal. Making further use of the symmetric properties of the circulant matrices and defining a dot-wise product operator also, the term $X^H X$, and u can be represented in compact form as:

$$X^H X = F diag(\hat{\boldsymbol{x}}^* \odot \hat{\boldsymbol{x}}) F^H \tag{I.2.5}$$

Here the dot product between the brackets represents the autocorrelation of input $\boldsymbol{x}$ in the Fourier domain. Typically autocorrelation of a time signal represents the shifted time lags with itself, applied to image data; this represents the spatial 'lag'. Combining all of the above, the minimisation problem can be formulated as:

$$\hat{\boldsymbol{w}} = \frac{\hat{a}^* \odot \hat{z}}{\hat{a}^* \odot \hat{a} + \lambda} \tag{I.2.6}$$

A final step in the methodology of the KCF is the introduction of kernel functions in the linear ridge regression by applying the so-called 'kernel trick'. In many applications, data is much easier to separate in higher-dimensional space. The kernel trick efficiently evaluates the dot product in a higher dimensional space without the need for costly transformations of the input vectors. Following the methodology described in [7], the Kernelised formulation for coefficient updates gives:

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{\boldsymbol{y}}}{\hat{\boldsymbol{k}^{xx}} + \lambda} \tag{I.2.7}$$

Here the hat terms similarly represent the vectors in the Fourier domain, $\boldsymbol{k}^{xx}$ represents the first row in the kernel matrix defined as $K = C(k^{xx})$ and $\hat{\boldsymbol{\alpha}}$ represents the vector of the coefficients α_i in this higher dimensional space, similar to the formulation of weights in eq. I.2.3:

$$\boldsymbol{\alpha} = (K + \lambda I)^{-1}\boldsymbol{y} \tag{I.2.8}$$

To allow detection of the target object, the regression function $f(\boldsymbol{z})$ needs to be evaluated at several image locations or candidate patches. The candidate patches can be constructed by adopting the cyclic shift model, allowing the definition of the kernel matrix as $K^z = C(\boldsymbol{k}^{kz})$. Here, $\boldsymbol{x}$ is the base sample, $\boldsymbol{z}$ is the base patch, and k^{xz} is the kernel correlation of the two vectors or the relative shift of two vectors. Evaluating the regression function using circular shifts is effectively a spatial filtering operation. This operation can be most efficiently performed in the Fourier domain. The regression function is then represented as a linear combination of kernel values k^{xz} and learned coefficients $\boldsymbol{\alpha}$:

$$\hat{f}(z) = \hat{k}^{xz} \odot \hat{\alpha} \tag{I.2.9}$$

Various kernel functions are possible. To relax the computational effort KCF proposes an Radial Basis Function (RBF) Gaussian kernel function $k(x, x') = exp(-\frac{1}{\sigma^2||x-x'||^2})$ arriving to the following kernel correlation form and $O(nlogn)$ computational time:

$$\boldsymbol{k}^{xx'} = exp(-\frac{1}{\sigma^2||x||^2 + ||x'||^2 - 2\mathcal{F}^-1(\hat{x}^* \odot \hat{x}')} \tag{I.2.10}$$

For each future frame, the coefficients learned in the previous frame can be updated by linear integration with new parameters. This allows building a memory into the model that is controllable by the regularisation parameter $\lambda \in [0, 1]$:

$$\hat{\boldsymbol{\alpha}}(t) = \lambda\hat{\boldsymbol{\alpha}} + (1 - \lambda)\hat{\boldsymbol{\alpha}}(t - 1) \tag{I.2.11}$$

I.2.2.2. EKF and AEKF: State and Parameter Estimation

A purely visual filter does not consider the dynamics of the object subject to tracking and therefore does not show robustness against occlusions and fast-moving objects. This section investigates how Extended and Augmented Kalman Filters can be coupled to the visual filter for state and parameter estimation.

Simplified Dynamic Model

The simplest model of a Kalman Filter, which can be used in unison with visual tracking, has linear motion dynamics. Since the wing typically exhibits oscillatory motion under aerodynamic input loads, the performance and the robustness of the visual tracking can be improved by extending the simplified dynamics of a moving linear particle to a typical spring-mass damper system. This robustness is essential when the LEDs fail for more extended periods (longer than half a period of one typical oscillation), and momentarily, no observation is available of the visual markers. The clamped wing will oscillate around its equilibrium under external excitation, structurally representing a clamped beam model beam. If the intervals of led loss are small, the moving particle is robust enough to interpolate the motion. However, it will not be able to capture the oscillations dynamics. This

oscillatory dynamics can be captured by a simple linear 1-DOF second order mass damper system, where we look at displacement $y(t)$ of a point mass. The general differential equation is given as:

$$\ddot{y}(t) = -\frac{c}{m}\dot{y}(t) - \frac{k}{m}y(t) \tag{I.2.12}$$

In the state space form, we have the following:

$$\frac{d}{dt}\begin{bmatrix} y_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix}\begin{bmatrix} y_k \\ \dot{y}_k \end{bmatrix} \tag{I.2.13}$$

It must be noted that in this form, we consider the homogeneous form for two reasons: (i) gust (external) input cannot be measured directly, and (ii) the aim is to maintain the Kalman Filter as model-free as possible. The gust input can be modelled as initial displacement in this Kalman Filter form.

DISCRETE KF FORMULATION

To deal with time intervals of image capture, $h = \Delta t$, a discrete Kalman Filter form must be formulated. Using a typical Euler integration given in this form of the state x:

$$\frac{d(x(t)}{dt} = \dot{x}(t) = f'(x(t)) = \frac{x(t+h) - x(t)}{h} \tag{I.2.14}$$

Rewriting gives the definition of the next time step:

$$x(t+h) = x(t) + f'(x(t))h \tag{I.2.15}$$

Now in discrete form using $k = t + h$:

$$x_k = x_{k-1} + \dot{x}_{k-1}h \tag{I.2.16}$$

The Kalman Filter contains the states $\bar{x}_k = \begin{bmatrix} y_k & \dot{y}_k \end{bmatrix}^T$. Each state of the Kalman Filter can be rewritten in a discrete recursive form. The states of the Kalman Filter are:

$$y_k = y_{k-1} + \dot{x}_{k-1}h \tag{I.2.17}$$

$$\dot{y}_k = \dot{y}_{k-1} + \ddot{y}_{k-1}h \tag{I.2.18}$$

Using the expression for the acceleration, the velocity can be rewritten into:

$$\dot{y}_k = -k/m \cdot y_{k-1} - (1 - c/m \cdot h) \cdot \dot{y}_{k-1} \tag{I.2.19}$$

Now the following system state matrix is obtained:

$$\bar{\mathbf{x}}_k = \begin{bmatrix} y_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} 1 & h \\ -k/m & -(1 - c/m \cdot h) \end{bmatrix} \tag{I.2.20}$$

Where the output is:

$$\bar{\mathbf{z}}_k = \begin{bmatrix} 1 & 0 \end{bmatrix}\bar{\mathbf{x}}_k \tag{I.2.21}$$

While changing the model to this oscillator gives more robustness against LED marker loss, it introduces other complications:

- The model has now become slightly more complex, and model parameters m, c, k need to be estimated.
- Additionally, we can no longer work with arbitrary time steps, and the actual time step between the image frames is required. This time step difference must set the discrete sampling, dt of the Kalman Filter at each iteration of the tracker algorithm.
- Furthermore, transformation is now required to go from x, y frame coordinates to the frame of reference for the modelled oscillator system.

However, the points mentioned above implicitly imply that the more complex model approach may no longer be model free. To overcome this, the proposed approach is to make the model nonlinear (time-varying in c, k and m parameters) and use AEKF to estimate these model parameters. The way this can be achieved, is to augment the previous state vector $\mathbf{x}_{k_{sys}}$ with unknown model states $\mathbf{p}_{k_{model}}$. For full time-varying spring-mass-damper system this will become, $\bar{\mathbf{x}}_k = \begin{bmatrix}\bar{\mathbf{x}}_{k_{sys}} & \bar{\mathbf{p}}_{k_{model}}\end{bmatrix} = \begin{bmatrix}y_k & \dot{y}_k & K_k & c_k & m_k\end{bmatrix}^T$.

AEKF Formulation

To rewrite the model into an augmented form and perform parameter estimation, the system parameters can be made a function of time. The differential equation can take the following form:

$$\ddot{y}(t) = -\frac{c(t)}{m(t)}\dot{y}(t) - \frac{k(t)}{m(t)}y(t) \tag{I.2.22}$$

One commonly used assumption for the augmented model is to assume that the derivative of model parameters is zero: $\dot{\mathbf{p}}_k = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$. Now the following nonlinear system matrix is obtained:

$$\bar{\mathbf{x}}_k = \begin{bmatrix} y_k \\ \dot{y}_k \\ K_k \\ c_k \\ m_k \end{bmatrix} = \begin{bmatrix} y_{k-1} + \dot{x}_{k-1}h \\ -K_{k-1}/m_{k-1} \cdot y_{k-1} - (1 - c_{k-1}/m_{k-1}h) \cdot \dot{y}_{k-1} \\ K_{k-1} + 0 \cdot h \\ c_{k-1} + 0 \cdot h \\ m_{k-1} + 0 \cdot h \end{bmatrix} \tag{I.2.23}$$

In order to solve the AEKF model at each time step, the Jacobian of the system matrix w.r.t time-dependent states must be constructed by linearising the model at each time step:

$$\mathbf{J}(\bar{\mathbf{x}}_k) = \begin{bmatrix} 1 & h & 0 & 0 & 0 \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{I.2.24}$$

with:

$$J_{21} = -K_{k-1} \cdot m_{k-1}^{-1} \cdot h, \qquad J_{22} = 1 - c_{k-1} \cdot m_{k-1}^{-1} \cdot h, \qquad J_{23} = m_{k-1}^{-1} \cdot y_{k-1} \cdot h,$$
$$J_{24} = -m_{k-1}^{-1} \cdot \dot{y}_{k-1} \cdot h, \qquad J_{25} = m_{k-1}^{-2} \cdot c_{k-1} \cdot \dot{y}_{k-1} \cdot h - m_{k-1}^{-2} \cdot K_{k-1} \cdot y_{k-1}$$

The output of interest remains the measured displacement; however, the system parameters are now estimated at each time step as output.

I.2.2.3. IMAGE PROCESSING AND MOTION RECONSTRUCTION PIPELINE

This section describes the complete processing and reconstruction steps required for real-time tracking of the wing shape and motion.

CLUSTERING

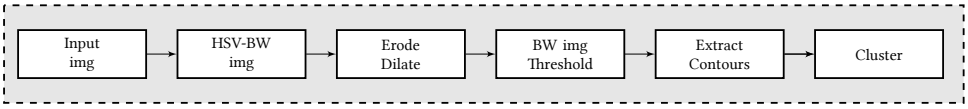


Figure I.2.3: Schematic of HSV filtering, such that a BW image is obtained for subsequent clustering.

To track visual markers, a processing pipeline is required. The initial step is to detect and cluster the markers correctly. For this purpose, a machine learning approach is developed. The method utilises DBSCAN and inverse formulation of DBSCAN proposed in the earlier study [9]. DBSCAN was found to be particularly suitable for the task due to its unsupervised nature, namely (i) minimum needed domain knowledge, (ii) ability to find clusters of varying size and (iii) ability to deal with noise. The image filtering pipeline for the clustering process is shown in Fig. I.2.3. An unsupervised clustering method was essential due to the periodic failure of the led markers installed on the wing. This is elaborated in Chapter I.1 and illustration in Fig. I.1.6.

RADIAL SORTING

A radial sorting algorithm is proposed in the processing routine after clustering operations to obtain a geometrical representation behind the detected clusters. The algorithm is initiated by finding the centroid $\bar{\mathbf{c}}_{cp}$ of the cluster centres (a cloud of points) $\mathbf{P}(x, y)$, then obtaining a radially sorted distribution, a so-called convex radial hull, $\mathbf{P}_{\theta hull}$, of n indices, such that the outline of the hull has continuous connectivity. The algorithm is discussed in detail in Chapter I.1, Sec. I.1.2.5

RECONSTRUCTION

A high-level overview of the entire tracking pipeline is shown in Fig. I.2.4.

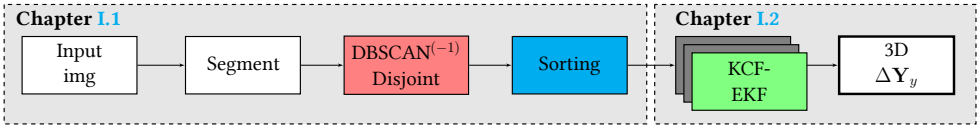


Figure I.2.4: High-level overview of the full 3D tracking and reconstruction pipeline.

The red block is the clustering algorithm (DBSCAN/DBSCAN^{-1}/Disjoint) implemented in this study. The green blocks represent an independent tracking filter and Kalman Filter duos (KCF-AEKF) that run in parallel to keep track of markers through a sequence

of images. The output is the displacement of the marker in x, y pixel coordinates of the frame. The reconstruction is the final step that relates the displacements of corresponding markers in two frames and reconstructs the 3D displacement. The reconstruction process can be inferred from the schematic of the camera setup shown in Fig. I.2.5. Further details regarding the 3D reconstruction can be found in a previous study by Mkhoyan et al. [10], and [11].

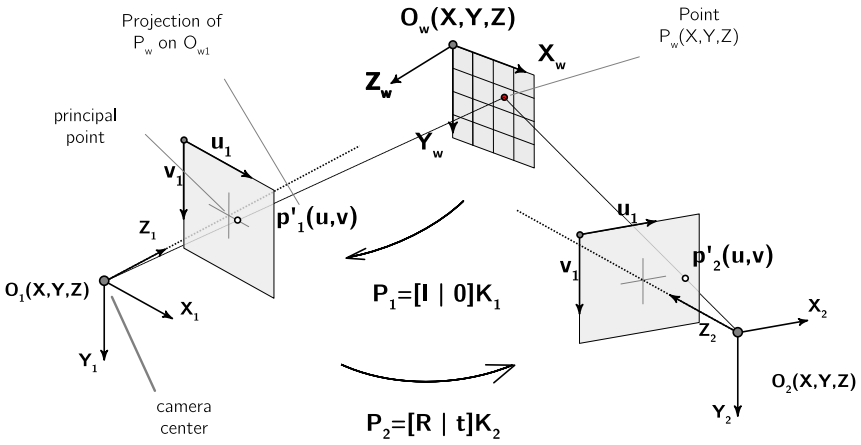


Figure I.2.5: Schematic of the stereo camera setup and the coordinate systems for 3D reconstruction.

I.2.3. Experimental Setup

To collect the experimental data and, subsequently, test and assess the proposed methods, a flexible wing was equipped with an array of visible (active) LED markers and subjected to gust excitation. The study is the continuation of a previous study [9] on smart sensing methods for the control of flexible aircraft.

I.2.3.1. Apparatus

The experiment was conducted in the Open Jet Facility (OJF) located at the Aerospace Engineering faculty of the Delft University of Technology [12]. The OJF is a closed-circuit low-speed wind tunnel with an octagonal test section of 285x285 cm^2. The wind-tunnel facility shown in Fig. I.2.6 was equipped with a gust generator [13] composed of two gust vanes actuated in parallel. The gust vanes allowed generating discrete (1-cosine), harmonic and sweep signals, resulting in wing gusts on top of nominal wind tunnel free stream velocity. The gust vanes were operated at maximum of 5-7 Hz for gust vane angle of $\alpha_g <= \pm 15°$, and 10-15 Hz for vane angles $\alpha_g <= \pm 10°$. The freestream velocity in the wind tunnel was maintained by a 500 KW electric engine. The maximum flow velocity available in the wind tunnel is 35 m/s; however, the theoretical performance limit is around 30 m/s.

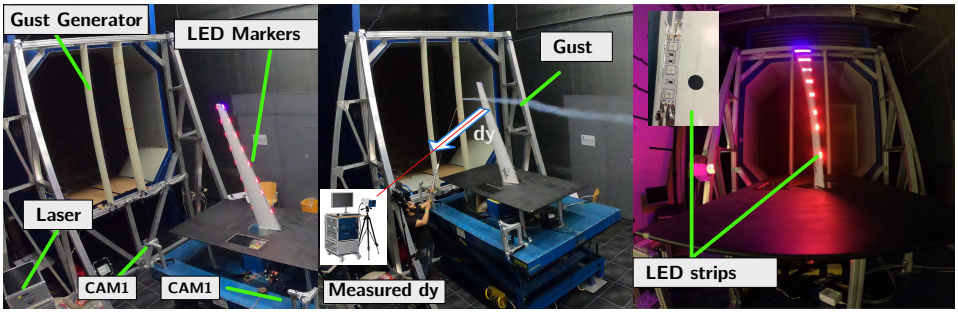


Figure I.2.6: The Open Jet Facility (OJF) [12] with the gust generator mounted in front of the test section. Wing flexing with LED strips installed on the right.

To measure the deflections resulting from the gust excitation, a Polytech PSV-500 laser vibrometer system was used with a resolution (RMS) of 200 µm/s [14]. In total 8 markers were measured by the system as shown in Fig. I.1.12a in Sec. I.1.3.3.

I.2.3.2. Wing Model

The wing used in the experiment was a forward-swept tapered wing, built of glass fibre reinforced epoxy material, referred to as the *Allegra wing*. The design of the wing allows for large tip displacements, up to 20% for 10° of AoA and 50 m/s flow velocity [15]. The wing was clamped on one side on a sturdy table under a fixed angle of attack of 4°. Detailed information about the wing can be found in [16].

The wing was equipped with 16 LED markers. Each LED marker consisted of 3 sub-LED units, providing three distinct bright light sources per marker. The experiment used a 1-cosine gust signal and a frequency sweep signal.

I.2.3.3. Experimental Conditions

For all the experimental condition cases, the experimental setup was kept constant. The wind tunnel test consisted of two experimental conditions, (i) static and (ii) dynamic, two camera conditions (perfect, dark) and two wing configurations (no mass, mass). The latter two wing configurations were introduced to investigate the effect of wing mass matrix and wing inertia change on tracking performance and the parameter estimation of the mass parameters in the AEKF oscillator model.

The purpose of the camera conditions was to assess the robustness of visual tracking in the presence of external visual disturbances. The camera conditions were designed to simulate realistic lighting conditions such that the feasibility of the method in-flight could be evaluated. For all three conditions, the illuminance of the room was to be measured to establish a baseline sensitivity for light intensity and tracking performance.

An overview of independent and dependent variables for the wing configuration and camera is given in table I.2.1.

Table I.2.1: Parameters of the wing and camera setup in the wind tunnel experiment.

	IV's (controlled)		DV's (measured)	
	parameter	unit	parameter	unit
Camera	I_{led} (led intensity)	[W]	I_{room} (illuminance room)	[lux]
	$N_{trackers}$	-	H (camera frame)	[pix]
Wing	α AoA	[deg]	$\delta \mathbf{Y}_x$ (displacement-x)	[m]
	v_o (flow velocity)	[m/s]	$\Delta \mathbf{Y}_y$ (displacement-y)	[m]
	(wing tip mass)	[kg]	error displacement	[m]
	f_g (gust frequency.)	[Hz]		
	d_g (gust vane angle.)	[deg]		

Camera Conditions

Two visibility conditions were used, C1 *dark* conditions (night visibility, low background light corresponding) and C2 *bright* condition (daylight visibility, high background lighting). The visibility was selected to study the effect of tracking and sensitivity of HSV filtering.

Motion Conditions

The motion conditions are composed of two cases, static and dynamic. In both cases, each condition was repeated for different camera configurations. The dynamic motion conditions were selected to produce a high dynamic response from the wing and thus more pixel activity in the image. The gust vane frequency of 5 Hz was close to the wing's natural frequency at the given mass configuration.

The objective of the static case was to assess the ability to (i) extract absolute displacements and (ii) monitor any steady-state error (bias) developed during the measurement runs in tracking the position of the markers.

In the dynamic case, the aim was to assess the (i) robustness of the tracking and (ii) monitor the measurement bias (drift) developed over time. The robustness indicator is the ability of the tracker to maintain the tracked marker for high-frequency oscillations. The aim was to cover the aerodynamic behaviour of the wing for typical free stream velocities and angles of attack. The test matrix of the experimental motion conditions is given in table I.2.2.

Table I.2.2: Test matrix of the experimental motion conditions.

Condition	signal	V_0 [m/s]	AoA [deg]	Gust f. [Hz]	Gust v.a. [deg]	Camera c.
static		[5,10,20,25,30]	[2]	-	-	[C1,C2]
dynamic (gust)		[10,30]	[2]	[1,2,3,4,5,5.6]	[5,6,7,8,9,10]	[C1,C2]
dynamic (sweep)		[10,30]	[2]	∞	[5,10]	[C1,C2]

I.2.3.4. Visual Tracking Setup

An overview of the hardware used for dataset collection is shown in Fig. I.1.12 in Sec. I.1.3.3. The dataset was recorded with two GigE acA1300-75gc Basler ethernet cameras with 1300 CMOS 1.3 megapixel (280×1024 pixels) sensor [17]. The cameras were equipped with Computar 12 mm F1.4 2/3" P IRIS lenses [18] and were positioned in a stereo setup to observe the markers from two viewpoints. The resulting image was cropped to 1088×600 pixels and streamed in 3 channel RGB format synchronously via real-time PTP triggering protocol over the ethernet. A PoE smart switch GS110TP from NETGEAR provided both the power, 3.5 W (per camera unit), as well as the GigE capability to stream the images up to 140 FPS.

An embedded computing system delivered the processing power and image capture from NVIDIA, the Jetson TX2, equipped with NVIDIA Pascal architecture with 256 NVIDIA CUDA cores and 1.3 TFLOPS (FP16), Dual-core Denver 2 64-bit CPU and quad-core ARM A57 complex [19]. The Jetson TX2 is designed for embedded applications using Artificial Intelligence (AI) and Computer Vision (CV) and operates on Ubuntu 16.04 LTS allowing flexibility in code deployment. The application developed for this Chapter was programmed in C++ and deployed on the device. For the development the Basler C++ Pylon API [17] and OpenCV open-source computer vision library [20]. The image and tracking data were extracted and plotted using the OpenCV-Matlab parsing interface repository tmkhoyan/cvyamlParser [16].

Code development, testing and assessment were done using standard Dell Optiplex 7400 and 2.3 GHz Intel Core i5 16G MacBook and the Jetson TX2. A part of the code, dataset and tools developed in the scope of the study are available under the repositories tmkhoyan/adaptiveClusteringTracker [9, 11] and tmkhoyan/parallelTrackingTBB [10, 21].

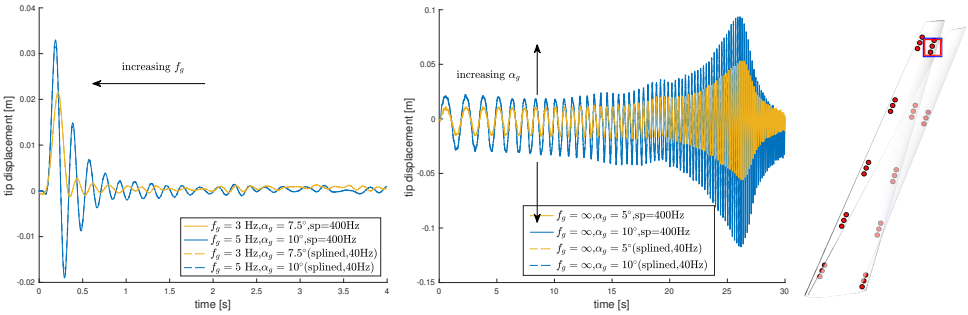
I.2.4. Results and Discussion

The results of the OJF wind tunnel test conducted with the flexible and the gust generator are discussed in this section.

I.2.4.1. Measured Wing Response

In Fig. I.2.7 the responses to 1-cosine gust and sweep input signals are shown. The time history signals correspond to the measurements taken at marker ID 1. The labelling and notation of the marker IDs for the vibrometer measurement system are shown in Fig. I.1.12a. Figures I.2.7a andI.2.7b show the wing's response to a single gust input and a sweep signal, respectively. The solid curves correspond to the measurement by the laser vibrometer sampled at 400 Hz; the dotted line is a spline model of this response sampled at the capture intervals by the leading edge camera. This spline model is required to obtain synchronised measurement points between the laser vibrometer data and the image sequences for comparison. The camera images were collected at approximately 40 Hz, with the Nyquist frequency well above the expected resonance frequency of the wing of ≈ 5 Hz.

The wing deflection at static motion conditions (various free stream velocities) is shown in Fig. I.2.8b shows the baseline static deflections obtained at varying free stream velocities. In Fig. I.2.8a the dynamic response of the wing as a result of the gust excitation is shown in a spatial and time-domain representation.



(a) 1-cosine gust input signal for varying gust vane angles and gust frequencies at a flow velocity of $V = 30$ m/s.

(b) Sweep input signal with varying gust vane angles at flow velocity of $V = 30$ m/s.

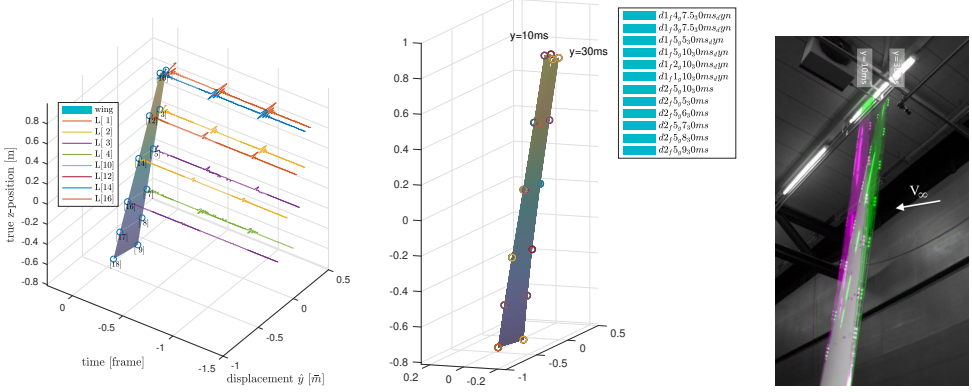
(c) Tracker location.

Figure I.2.7: The laser vibrometer measurement (solid line), of the tip displacement of marker ID 1, sampled and splined at capture intervals of the leading edge at ≈ 40 Hz (dotted line).

I.2.4.2. Gust Parameter Sensitivity and Wing Response Analysis

The wing response to various gust inputs was studied to better understand the requirements and the needed performance of the visual tracking. Here, the sensitivity of wingtip response (marker ID1) to 1-cosine gust input (α_g,f_g,V) parameters was measured.

Observing Fig. I.2.7a the effect of the change of input gust frequency is seen by comparing the high frequency (blue) to low frequency (yellow). At higher frequencies, above 3 Hz, the wing response resembles more an impulse response, while at lower frequencies, it resembles more a 1-cosine input signal. Fig. I.2.9a shows the full range of frequencies (1-5 Hz). This shift in response type appears to happen between the 3 Hz (yellow) and 4 Hz (red) lines. This can be explained as for narrower 1-cosine gust inputs, the response



(a) 3D view and marker tracking in time-domain.

(b) Baseline static deflections at varying V.

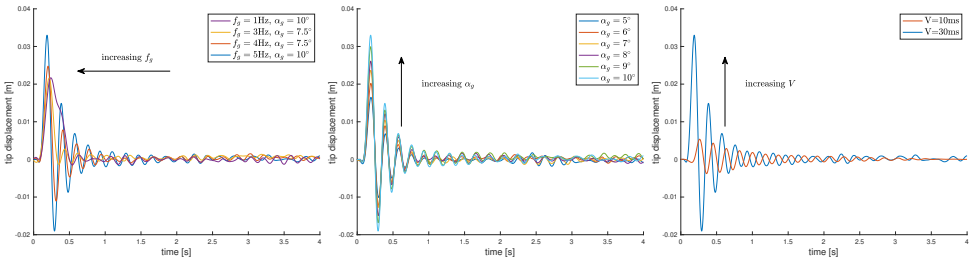
(c) Wind direction.

Figure I.2.8: Spatial and time representation of the series of wing deflection under static and dynamic loads.

tends to attain the characteristics of an impulse, and the energy is absorbed in a narrower time span by the wing.

In contrast, below the 3 Hz, the response is spread over a longer time. The purple line in Fig. I.2.9b shows this larger spread between 0-0.5 seconds. This observation indicates that the higher frequency responses are more critical for visual tracking above 3 Hz. Therefore, the 5 Hz gust performance is at the centre of the following discussions.

Further observations can be made regarding the sensitivity of the gust vane angle and the free stream velocity. Fig. I.2.9b shows the responses for varying gust vane angles of the gust generator ranging from 5 to 10 degrees (blue to cyan lines), while the free stream velocity is kept at 30 ms. Higher gust vane angles generate larger amplitude cosine waves corresponding to more significant input energy and higher lift. As seen from Fig. I.2.9b, this results in larger amplitudes of the wing response. Similarly, larger free stream velocities result in higher amplitudes. Fig. I.2.9c shows how the response amplitude is increased by over six-fold, from approximately 0.005 to 0.032 meters, between 10 ms (red) and 30 ms (blue). Both of these motion responses are relevant for visual tracking, as higher amplitudes in a shorter period challenge the capture and processing frame rate of the visual tracking, and lower amplitudes, the sensitivity threshold for pixel activity, directly depends on the image resolution. However, it must be noted that larger amplitudes are more relevant for controller feedback. The differences between the gust vane angles are impacting on the response amplitude. Therefore, mainly the gust vane angle of 10 degrees is discussed.



(a) Varying frequency f_g at constant V of 30 ms.

(b) Varying gust vane angle α_g at constant V of 30ms and f_g of 5 Hz

(c) Varying free stream velocity V at constant α_g of 10 ° and f_g 5Hz.

Figure I.2.9: Effect of gust input parameters on the measured wing response.

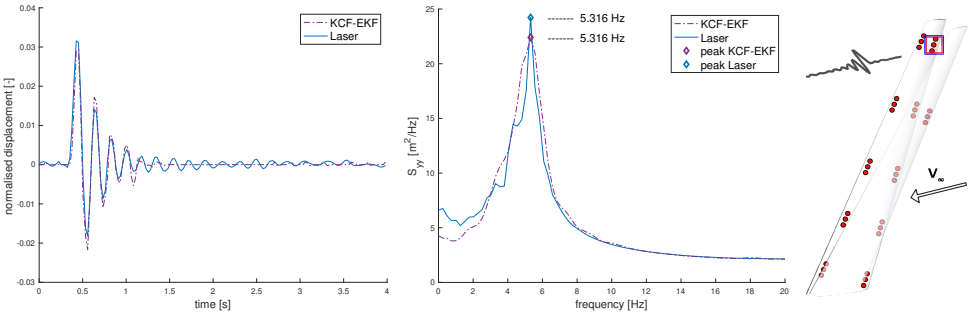
Regarding the sweep input, it is observed that the increasing gust vane angle results in a larger amplitude. This is shown by the wider amplitude band of the 10-degree signal (blue) versus the 5-degree signal (yellow). The lower amplitude signal (yellow) is expected to be more challenging for visual tracking as less activity is expected in the subsequent image frames.

I.2.4.3. Motion Reconstruction and Analysis in Frequency and Time-Domain

To quantify the measured wing response and the tracked wing motion from the image sequences, comparisons were made in terms of displacements and the frequency content in Fig. I.2.10. The laser vibrometer measured response (in meters) was reconstructed with

the KCF-EKF (green) pipeline I.2.4. The camera setup was calibrated to obtain the reconstructed displacements, and the corresponding KCF-EKF point pairs from two image streams were triangulated with the Direct Linear Transformation (DLT) method. The details of the triangulation approach are given in [10, 21]. The obtained 3D coordinates were then transformed to align the laser vibrometer measurement. Similar to laser vibrometer measurement, the baseline static displacement was subtracted to obtain the transient displacement $\Delta \mathbf{Y}_y$ as indicated by the last block in the diagram I.2.4.

Observing the reconstructed displacement of the tip marker in Fig. I.2.10a, a good agreement with the laser measurements is found in terms of the phase of the response. The reconstruction seems to overshoot the response at the amplitude peaks. In particular, after the first peak, the difference between the laser and reconstruction is approximately 3.5 mm. In the first peak, the difference is significantly lower, below 2 mm. This difference can be attributed to three reasons, namely, (i) the splining of the laser reference measurement tends to undershoot the peak due to a lower sampling rate, (ii) transformation between laser measurement and absolute reconstructed coordinates and (iii) stereo calibration errors. The latter factor plays a significant role in the tracking quality and requires carefully calibrating the entire volume of the 3D space where the motion takes place. In the current case, the calibration could only be performed when the wing was already installed, which prevented the cover of the spatial domain of the markers. Furthermore, errors in measurement between the camera setup's orientation and location to the laser vibrometer's reference system also play a role in the comparison. A fourth reason is the smoothening of the peaks by the Kalman-filtering and the lag introduced. However, these can be more easily adjusted by tuning the Kalman Filter parameters with the KCF-AEKF approach to rely more on the KCF tracker measurement. It must be noted that this, however, would also reduce the robustness of the tracking to occlusions; hence a trade-off must be made.



(a) Comparison response of marker ID 1. (b) Auto-PSD of the response signal at marker ID 1. (c) Tracker location.

Figure I.2.10: Comparison of the response and the frequency spectrum for laser vibrometer versus image tracking pipelines with DBSCAN and the Disjoint-set data structure for run R1 and marker ID 3.

Alternatively, the good agreement of the phase of the reconstructed motion can be further analysed by employing a frequency domain analysis. Here, the measurements were compared in terms of the measured output's power spectral densities (PSD). The main objective of this analysis was to understand whether the image tracking methods could correctly identify the frequency content of the measured signal compared to the

reference measurement provided by the laser vibrometer. In this context, the aim was not to extract the system's frequency response function; hence no cross-correlation was performed of the input to the output signal, and the power spectrum of the output alone provided sufficient ground for comparison.

The auto-PSD of the output signal, S_{yy}, was calculated according to the following definition:

$$S_{yy}(\omega) = \int_{-\infty}^{\infty} R_{yy}(\tau) e^{-j\omega\tau} d\tau \tag{I.2.25}$$

The integral in the expression is the Fourier transform of the auto-correlation function R_{yy} of the output signal (marker displacement). Figure I.2.10b shows the frequency content of the image sequence corresponding to the responses of marker ID 1. As can be observed from Fig. I.2.10b, the tracking method can provide a good estimate of the first resonant frequency of the wing. The result shows that both peaks of the spectral densities align at 5.316 Hz, and despite underestimation of the amplitude peaks, the resonance region is captured well with the reconstructed signal. In general, the time and frequency domain analysis results suggest that the oscillatory motion of the wing can be well captured and reconstructed with the relatively low-resolution cameras (1.3 megapixels) at capturing rate of 40 Hz. The accuracy can be further improved by higher resolution cameras and a higher capture rate.

I.2.5. Conclusions and Recommendations

This Chapter proposed a method for robust visual tracking and reconstruction of flexible wing motion for controller feedback. The method consisted of a tracker pair composed of a purely visual filter, a high-speed Kernelised Correlation Filter (KCF) [7], paired with an Augmented Extended Kalman Filter (AEKF), allowing an adaptive estimate of the states and parameters of the system exhibiting oscillatory motion. The method consisting of KCF-EKF, without Kalman Filter augmentation, was validated experimentally on a real-time image stream of a very flexible wing subjected to gust excitation in the OJF (Open Jet Facility) at the Delft University of Technology. From the measured experimental data obtained with the laser vibrometer system, the effect of gust frequency, the gust vane angle and the free stream velocity were studied on the wing response to demonstrate the requirements of the visual tracking and reconstruction. The reconstructed motion of the wingtip markers from KCF-EKF tracking showed good agreement with the reference measurement provided by the vibrometer. Frequency and time-domain analysis and data comparison showed that the resonance peak could be captured well with the reconstructed response from the visual tracking. The results suggested that the oscillatory motion of the wing can be well captured and reconstructed with relatively low-resolution cameras (1.3 megapixels) and a non-invasive sensing system. The low resolution allows us to reach higher capture rates for better reconstruction accuracy. A further study is planned to analyse the performance of the Extended and Augmented Kalman Filter for varying motion conditions under the presence of uncertainty in the visual information in the form of marker failure.

References

[1] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive State Estimation and Real-time Tracking of Aeroelastic Wings With Augmented Kalman Filter and Kernelized Correlation Filter," in *AIAA Scitech 2021 Forum*, p. 0666, 2021.

[2] A. W. Burner and T. Liu, "Videogrammetric model deformation measurement technique," *Journal of Aircraft*, vol. 38, no. 4, pp. 745–754, 2001.

[3] P. I. Corke, "Visual Control of Robot Manipulators – a Review," in *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, pp. 1–31, World Scientific, 1993.

[4] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern recognition letters*, vol. 34, no. 1, pp. 3–19, 2013.

[5] A. N. Belbachir, *Smart cameras*, vol. 2. Springer, 2010.

[6] T. Mkhoyan, N. R. Thakrar, R. De Breuker, and J. Sodja, "Design of a Smart Morphing Wing Using Integrated and Distributed Trailing Edge Camber Morphing," in *ASME 2020 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, pp. –, American Society of Mechanical Engineers, sep 2020.

[7] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 583–596, mar 2015.

[8] R. M. Gray, "Toeplitz and Circulant Matrices: A Review," *Foundations and Trends® in Communications and Information Theory*, vol. 2, no. 3, pp. 155–239, 2005.

[9] T. Mkhoyan, "tmkhoyan/adaptiveClusteringTracker: Initial public release," 2019.

[10] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Parallel Real-Time Tracking and 3D Reconstruction with TBB for Intelligent Control and Smart Sensing Framework," in *AIAA Scitech 2020 Forum*, p. 2252, American Institute of Aeronautics and Astronautics (AIAA), jan 2020.

[11] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive Real-Time Clustering Method for Dynamic Visual Tracking of Very Flexible Wings," *Journal of Aerospace Information Systems*, vol. 18, pp. 58–79, jan 2021.

[12] K. Jongkind, A. Falkmann, and H. van der Veer, "Open Jet Facility," 2020.

[13] P. Lancelot, J. Sodja, and R. De Breuker, "Investigation of the unsteady flow over a wing under gust excitation," *17th International Forum on Aeroelasticity and Structural Dynamics, IFASD 2017*, vol. 2017-June, 2017.

[14] Polytec, "Polytec Single-Point Vibrometers," 2020.

[15] M. Ritter, Y. M. Meddaikar, and J. K. Dillinger, "Static and dynamic aeroelastic validation of a flexible forward swept composite wing," in *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2017*, p. 0637, 2017.

[16] T. Mkhoyan, "tmkhoyan/cvyamlParser: Initial public release," nov 2019.

[17] Basler AG, "Basler ace acA1300-30gm - Area Scan Camera," 2019.

[18] Computar, "Computar Machine Vision Lens Catalog," tech. rep., Computar, 2017.

[19] "Embedded Computer Vision Real Time? Nvidia Jetson TX2 + VisionWorks toolkit - Myzhar's MyzharBot and more...," 2020.

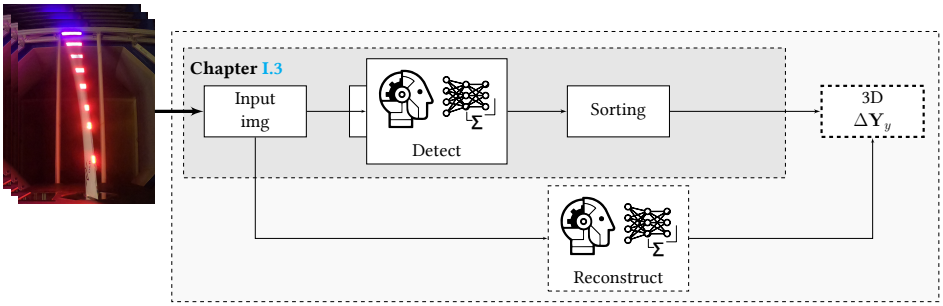
[20] G. Bradski, "The OpenCV Library," *Dr Dobbs Journal of Software Tools*, vol. 25, pp. 120–125, 2000.

[21] T. Mkhoyan, "tmkhoyan/parallelTrackingTBB: initial release," dec 2019.

I.3

Vision-Based Deep Learning Methods

In previous Chapters, a real-time wing shape reconstruction method was presented consisting of the various image processing steps. This study elaborates on the processing steps involved with the sorting and orientation of the detected markers using a traditional and a deep learning-based approach. In this Chapter, a computation of 2-axis reflectional symmetry from the based geometrical relationship of contour points was performed using the proposed GeConv algorithm. The algorithm is tested on a 1000+ image dataset, initialised with rotation, from a wind tunnel and flight test experiment as part of a more extensive study to apply real-time visual tracking of flexible wing structures. Furthermore, an alternative approach is presented using Deep Learning. Here a Deep Convolutional Neural Network is trained to detect the orientation angle of extracted contour shape of the wing. Both approaches show excellent results in detecting symmetry and orientation [1].



This Chapter is based on the following peer-reviewed conference paper:
T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Fast Symmetry Detection with Deep Learning and GeConv," in *IEEE RAS 2019 International Summer School on Deep Learning for Robot Vision*, IEEE, 2019

In many computer vision applications, knowing the object's orientation relative to its symmetry axis is relevant. While humans are generally very good at evaluating objects' reflectional symmetry and relative orientation, constructing a robust symmetry detection pipeline working out of the box is still challenging in computer vision. This is particularly true for irregular objects with skewed geometry and irregular shapes. This Chapter presents two symmetry and object orientation detection approaches based on traditional computer vision and Deep Learning. First, an algorithm is presented that allows fast computation of 2-axis reflectional symmetry of contour points and the definition of a radial convex hull, Geometrical Reflectional Symmetry Detector (GeConv). The algorithm is tested on a 1000+ image dataset from the wind tunnel and flight test experiment. Furthermore, an alternative approach is presented using Deep Learning. Here a 2D Conventional Neural Network is trained to detect the orientation angle of extracted contour shape of the wing. Both approaches show excellent results in detecting symmetry and orientation. The Neural Network can achieve an angle error of up to 0.05 degrees without needing a contour shape as input. However, it does not generalise easily to shapes deviating from the training set. The traditional computer vision approach excels at speed and is model-free; however, it requires a processing step to get the contour points. This study shows the potential for implementing deep learning-based methods for the wing shape reconstruction approach presented in the previous Chapters.

I.3.1. Introduction

The advancements in computer vision and the application of smartphones allow us to perform and process increasingly more complex image processing routines on better quality images. As the performance increases, the demand to do more complex tasks on more extensive data increases. One of the fundamental problems in computer vision has many applications in computer vision, ranging from shape recognition, shape completion, segmentation, mesh repair, and shape editing [2, 3, 3–5].

We as humans have developed a solid sensitivity for symmetry and geometric orientation of patterns allowing us to deduce the complex relationships between patterns and their orientation in the global coordinate system. Machines, however, require instructions to compute and compare parameters from the image data to deduce this information. In this study, two approaches are presented. One approach presents a fast, efficient algorithm, GeConv, to calculate the symmetry line of skewed image objects based on traditional computer vision using the so-called Rotational Geometric Convolution. The other approach mimics the human brain and utilises a Deep Convolutional Neural Network (DCNN) to train and classify the rotation of images from their initial state.

Aside from general use, these methods play an essential part in processing steps of wing shape reconstruction which was investigated in the previous Chapters, Chapters I.1 and I.2. In particular, the use of a deep learning-based approach for replacing steps or the entire process of wing shape reconstruction. Consequently, the approach was tested on an image sequence of a flexible wing with LED (light-emitting diode) markers undergoing oscillatory motion under gust excitations in the OJF (Open Jet Facility) wind tunnel at the Technical University of Delft. In the experiment, the gust generator is used developed for OJF in the previous study [6].

The main contributions of this Chapter are twofold. First, a comparison of conventional and deep learning-based approaches for detecting the symmetry and orientation of objects. Second, an investigation is made into the use of a deep learning-based approach for replacing steps or the entire wing shape reconstruction process.

This Chapter is structured as follows. The methodology is presented in Sec. I.3.2, where two approaches are proposed for detecting the detecting symmetry and orientation objects. In Sec. I.3.2.1 the GeConv algorithm is explained, which relies on the efficient use of traditional computer vision methods. The overview of the algorithm is presented in Sec. I.3.2.1. Next, in Sec. I.3.2.2, a rotation detector is presented based on raw pixel input and a Deep Convolutional Neural Network. The structure of the DCNN is explained in Sec. I.3.2.2 and the training approach in Sec. I.3.2.2. The experiment conducted to compare the two methods using a recorded wind tunnel dataset is explained in Sec. I.3.3. The results are discussed in Sec. I.3.4, specifically, in Sec. I.3.4.1 and Sec. I.3.4.2 for GeConv and Deep learning approach, respectively. Finally, the conclusions and recommendations are presented in Sec. I.3.5.

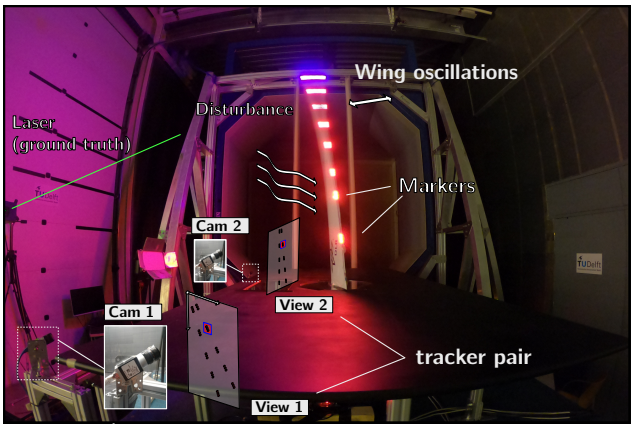


Figure I.3.1: Experimental setup with the wing facing the wind tunnel equipped with visual markers.

I.3.2. METHODOLOGY

This Section discusses the methodology of detecting symmetry and orientation objects intending to aid the wing shape reconstruction. Two approaches are proposed for detecting symmetry and orientation objects. In Section I.3.2.1 the GeConv algorithm is explained, which relies on the efficient use of traditional computer vision methods. The overview of the algorithm is presented in Sec. I.3.2.1. Next, in Sec. I.3.2.2, a rotation detector is presented based on raw pixel input and a Deep Convolutional Neural Network. The structure of the DCNN is explained in Sec. I.3.2.2 and the training approach in Sec. I.3.2.2.

I.3.2.1. GEOMETRIC CONVOLUTION: FAST SYMMETRY DETECTOR WITH ROTATIONAL GEOMETRIC CONVOLUTION

The fast Geometrical Reflectional Symmetry Detector uses machine learning and image processing to extract the contour points defining the detected shape or object. The main idea is to convolve a geometrical shape around a rotational centre defined as the centroid. This lends the name GeConv.

In the current study, the wing shape is extracted based on an outline of the wing shape defined by markers placed along the outline. The symmetry detector can also be used on shapes without distinct markers by including an alternative image segmentation filter in the routine. The pre-processed image is clustered with DBSCAN, after which a geometrical convolution is performed on the object's initial state. The symmetry detector can automatically sort the detected markers (object segment corners) based on their relative orientation concerning the centroid and the symmetry line. The geometrical convolution detector algorithm was written in C++, and the image processing was done using the C++ OpenCV library [7]

Frame Pre-Processing Filter

The pre-processing filter takes as input image 1088×600×3 pixels image and performs colour filtering operations to segment the object from the background. Since the wing is equipped with distinct markers, the outlines of the markers are filtered based on their colour properties. The colour filter allows light variations to filter the desired marker centres properly. The filter operates in parallel on each pixel of a full RGB image and filters the pixel's most distinct marker RGB value + uncertainty range. After this step, adaptive image thresholding is performed on the converted grey-scale image using the Otsu's [8] method based on pixel histogram values.

The latter method allows distinct segment regions based on the intensity of grey-scale value. The background and the object will typically show two distinct peaks, separated based on their statistical properties (mean and variance). This approach performs well when it is assumed that the histogram distribution is bimodal, meaning that segmented regions belong to the background or foreground class. The method shows limitations when the background contains noise or a small object area. The limitations can be easily overcome as the wing is equipped with distinct markers when no markers are available; the pre-processing filter must include an alternative segmentation approach to sort the objects of interest after Otsu's thresholding. The diagram of the preprocessing pipeline diagram and the result of the filtering are shown in Fig. I.3.2.

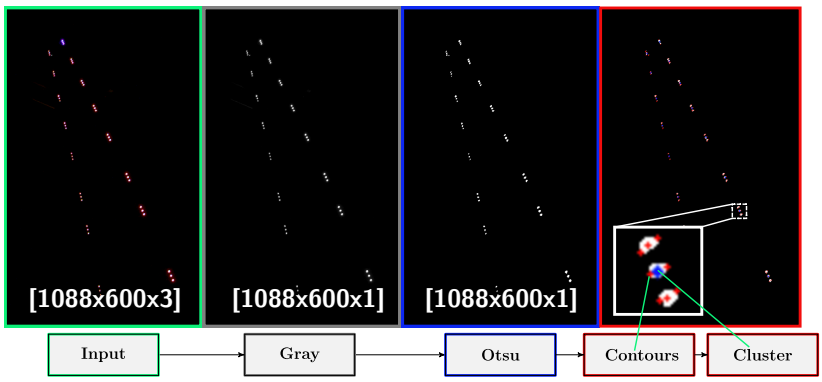


Figure I.3.2: Result of pre-processing filter and diagram. The images are from left to right, original, colour-filtered to greyscale, thresholding with Otsu and clustering with DBSCAN.

CLUSTERING WITH DBSCAN

A machine learning approach is used to tackle the problem of correctly detecting and clustering the markers. In the pre-processing, an unsupervised clustering method is used, DBSCAN [9]. This method is selected for its ability to classify noise and its unsupervised nature requiring minimum domain knowledge correctly. In pre-processing the given data, applying the unsupervised clustering method was crucial to ensure the method's robustness in the event of marker loss. Losing LED markers can occur when the LED circuit is under high strain during large wing oscillations. This is elaborated in Sec. I.1.2.3, Fig. I.1.6.

The main principle of DBCAN is to identify and separate regions of high density from low-density regions. At any given point, p, density is measured within a circular radius of ϵ. A dense region of radius ϵ from point p is a region that contains at least $MinPts$ number of points. The last two parameters are the main parameters of the algorithm. Given a database D the epsilon neighborhood, N_ϵ, of point p w.r.t. point q has the following form [9]:

$$N_\epsilon\left(p\right) = q \in \left|dist\left(p,q\right)\right. \tag{I.3.1}$$

This definition alone, when used naively, will fail to distinguish core points (points inside of the cluster), border points (points at the border of a cluster), and noise (a point not belonging to any cluster). The reason is that the ϵ neighbourhood of border points generally has much fewer points than the ϵ neighbourhood of a core point. The problem arises when the $MinPts$ parameter is set to a low value to include the border points, which can cause noise to be included in the cluster. To overcome this, DBSCAN introduces the concept of density reachability. A point is said to be *Directly Density Reachable* when the following two conditions hold:

$$p \in N_\epsilon\left(q\right) \tag{I.3.2}$$

$$\left|N_\epsilon\left(p\right)\right| \geq MinPts \quad \text{(core point condition)} \tag{I.3.3}$$

This condition thus sets a requirement for every point p in a cluster to be in the ϵ neighbourhood of another point in this cluster q. Additionally (ii) ϵ neighborhood of q, $N\left(q\right)$ must have a minimum of $MinPts$, classifying it as a core point.

The method introduces connectivity conditions for connecting N_ϵ of points and defines noise as a point not belonging to any cluster in dataset D under the given conditions (Density-Reachability and connectivity).

In Fig. I.1.6c the result is shown of the clustering operations both for DBSCAN scan (purple).

GECONV ALGORITHM OVERVIEW

The core of the algorithm relies on finding the centroid, $\bar{\mathbf{c}}_p$ of a cloud of points $\mathbf{P}(x,y)$, then obtaining a radially sorted distribution, a so-called convex radial hull $\mathbf{P}_{\theta hull}$, of n indices and convolve the geometry concerning its original orientation until the distance vectors from each point to the cluster centroid are aligned with the vertical and horizontal axis. The algorithm can also find custom symmetry at 45 or any given angle.

The steps are as follows. The algorithm takes as input an arbitrary indexed cloud of clusters centres or a point cloud $\mathbf{P}(x,y) \in \mathbb{R}^{2\times n}$. The centroid of $\mathbf{P}$, $\bar{\mathbf{c}}(x,y)$ is calculated to obtain the distance vector pointing towards the centroid. If the input is a continuous

shape, the centroid is sampled at the contours of the area, otherwise for a collection of n points:

$$\bar{\mathbf{c}}_p = \frac{1}{n}\sum_{i=1}^{n} p_i \text{ and,} \tag{I.3.4}$$

$$\mathbf{d}_{cp} = \mathbf{P} - \mathbf{c}_{cp} \tag{I.3.5}$$

The next angle is calculated for each distance vector in the collection, and the vector is radially sorted around $\bar{\mathbf{c}}_p$ in the given orientation to obtain a so-called convex radial hull $\mathbf{P}_{\theta hull}$:

$$\boldsymbol{\theta}_{cp} = \text{ARCTAN2}(\mathbf{d}_{cp}), \text{ where for each point,} \tag{I.3.6}$$

$$\text{ARCTAN2}(p)_i = \text{ARCTAN2}\left(\frac{p_y}{p_x}\right) \tag{I.3.7}$$

The sorted index of angles is obtained from $sort(\boldsymbol{\theta}_{cp})$, and the convex radial hull is obtained from sampling by this sorted index:

$$\mathbf{P}_{\theta hull} = sort(\mathbf{P}, sort(\boldsymbol{\theta}_{cp})) \tag{I.3.8}$$

This is required to outline points in a continuously connected area. The radial sorting is similar to what is used in gift wrapping algorithms such as Jarvis march [10]. However, the latter algorithm sorts the points iteratively along with its outer convex hull corner points by visiting point by point.

Now, while we have an idea about how the points are radially distributed in our local frame starting from index 0, the index is *floating* w.r.t. the geometry, and thus, we do not have an idea about how this starting index related to the geometrical properties (symmetry).

To find the symmetry lines, a convolution must be performed at discrete samples of θ (0-360) for a given chosen sampling step t_θ. Firstly the geometry of sampled points must be transformed using the rotation at each k^{th} θ sample around the centroid:

$$\mathbf{R} = \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{bmatrix} \tag{I.3.9}$$

The new geometry becomes:

$$\mathbf{P}_{\theta_k} = (\mathbf{R} \cdot (\boldsymbol{\theta}_{\mathbf{k-1}} - \boldsymbol{\theta}_{cp})^T)^T + \boldsymbol{\theta}_{cp} \tag{I.3.10}$$

Consequently, as with the initial step, the direction of the new distance vector is calculated at k^{th} step, $\mathbf{d}_{k_{cp}}$. Note that this vector will have the same magnitude, as the distance to the centroid will not change, but its orientation in the global coordinate frame will. The angle is calculated again for this vector, obtaining a new angle vector $\boldsymbol{\theta}_{k_{cp}}$. No sorting is required here. However, the crux now to obtain the symmetry line is to calculate the mean of the vector orientations. For vertical and horizontal symmetry, these must lie along with the 90-270 symmetry and 0-180 symmetry, respectively. Repeating this for all θ samples,

we obtain a θ_{Vsymm} and θ_{Hsymm} score at each sample step, and the symmetry will be found at the minimum of this across the sampling domain:

$$\theta_{Vsymm} = min(|mean(|\sin(|\boldsymbol{\theta}_{k_{cp}}|)|), \theta_{Vsymm}) \tag{I.3.11}$$
$$\theta_{Hsymm} = min(|mean(|\cos(|\boldsymbol{\theta}_{k_{cp}}|)|), \theta_{Hsymm}) \tag{I.3.12}$$

The score must be as close to 1 to obtain the symmetry for each axis. This process is shown in Fig. I.3.5.

An alternative option is to sort the histogram of these values, which is more in line with what is done with the surface gradient of the pixel in on image frames in this study [11]. However, the above two equations are far more efficient than collecting histograms along the image frame's x and y directions. The main contribution of this method is another addition. Moreover, the mean may not give the perfect symmetry score at the vertical or horizontal axes for skewed geometry. This is because the corner points of the skewed geometry, with a strongly deviating distance-vector direction $\mathbf{d}_{cp}$, will *dampen* out the symmetry detection peak. To account for this, the algorithm takes a statistical threshold parameter, z_{th}, to calculate a statistical mean or a strimmed mean. This will reject the outliers strongly deviating from the equation's mean. This parameter defines the skewness or the noisiness in the object's symmetry. For strongly skewed objects, the parameter will be below, otherwise high (retaining a more significant part of the $\boldsymbol{\theta}_{k_{cp}}$ vector). The equation now becomes:

$$\theta_{Vsymm} = min(|trimmean(|\sin(|\boldsymbol{\theta}_{k_{cp}}|)^{z_{th}}|), \theta_{Vsymm}) \tag{I.3.13}$$
$$\theta_{Hsymm} = min(|trimmean(|\cos(|\boldsymbol{\theta}_{k_{cp}}|)^{z_{th}}|), \theta_{Hsymm}) \tag{I.3.14}$$

Where the $trimmean$ of a collection of values in vector p is defined as the percentile given in by z_{th}.

I.3.2.2. Deep Learning Rotation Detector

The symmetry detection using the artificial intelligence approach was obtained by training a DCNN. The following sections describe the methodology and structure behind this approach.

DCNN Structure

The network was composed of a deep CNN architecture. The CNN takes as input (28×28×1) pixels and predicts the rotation angle (0-360) degrees. The input was chosen small since the CNN is expected to learn low-level features such as edges, and no abundant spatial pixel information was expected to be needed for the task. Furthermore, this made the learning and later inference much less computationally extensive, such that the learning could be done on a modern laptop/pc CPU in a matter of minutes. Similarly, the depth of the network was constrained to 2 layers since no high-level features needed to be earned. A general overview of the DCNN architecture and the feature extraction process is shown in Fig. I.3.3.

The entire structure is composed of an input layer of 28×28×1. Next, two convolution layers are used with kernel sizes of 3×3 and 64 convolutional filters. These are followed by a max-pooling layer with a pooling kernel of 2×2, reducing the output to half of the

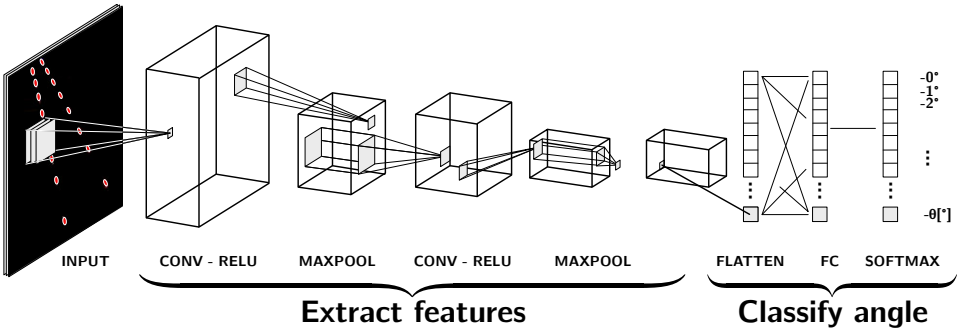


Figure I.3.3: General overview of the DCNN structure and feature extraction process.

previous layer's output. This is again followed by a dropout layer of 0.25 to help overcome overfitting during training and a flatter layer to reduce the input to one dimension. Finally, a fully connected layer, a final dropout layer, and a final fully connected layer with the number of outputs equal to the number of classes of rotation angles (0-360) follows. An overview of the layers in the DCNN structure is given in Sec. I.A1.2 in App. I.A1.3. The DCNN was constructed using Python, and Keras [12] with Tensorflow backend [13]. The image processing was done using the OpenCV library [7].

DCNN Problem Formulation and Training

The problem was treated as a classification problem, meaning the model's output was the probability between 0-1 of the predicted angle belonging to the class 0-360 integer range. Therefore, the output format was a vector of size $N = 360$ instead of a regression problem where the output is a single normalised value between 0-360. This was done such that no angle wrapping was required to deal with positive and negative differences between predicted and actual angle values.

The difference between the predicted and true value, Mean Absolute Error (MAE), was chosen for the loss function. Given as over a set of n predictions:

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y - \hat{y}_j| \tag{I.3.15}$$

As a type of loss function, a categorical cross-entropy was used suitable for classification tasks. The optimizer was a stochastic optimisation method Adam [14] with the angle error described above as the performance metric.

Frame Pre-Processing Filter

Pre-filtering was required to construct the suitable dataset as input for the CNN. The pre-processing filter takes as input same-size images (1088×600×3) pixels in RGB format, as with the previous pre-processing filter. However, the output needs to be reduced (28×28×1) pixels in greyscale format. First, the image was flattened to greyscale, reducing an image to 1 layer depth, after which the image was cropped and resized to a square

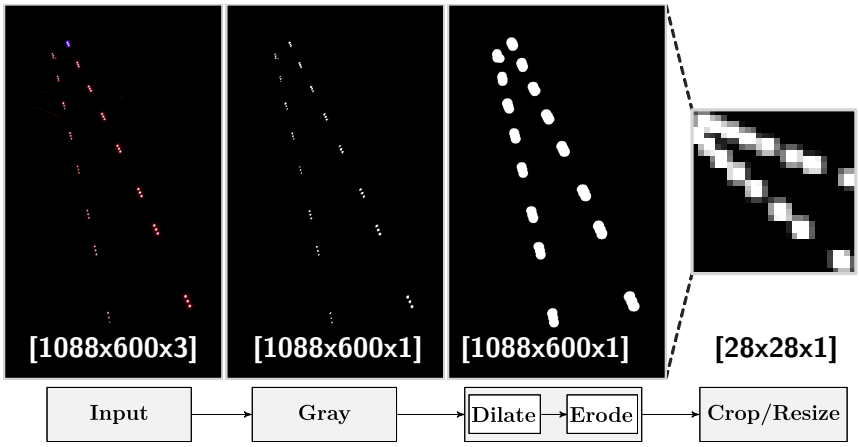


Figure I.3.4: Result of DCNN pre-processing filter. The images are from left to right, original, greyscale, thresholded after erode and dilate operations and after cropping and rotating.

shape. The resizing was done with linear interpolation. Since the input size is minimal, to retain distinctive edge features in the image after interpolation, erode and dilate operations were performed to increase the size of edges (marker features). The pre-processing pipeline and the result of the pre-processing are shown in the bottom row of Fig. I.3.4.

I.3.3. EXPERIMENTS

The experiments were performed on a 1000+ dataset of image sequences gathered from the wind tunnel test recordings. The data set was split into 0.8 ratios (training/evaluation). The images were rotated at an arbitrary angle from their initial setting as shown in Figs. I.3.2 and I.3.4 describing the pre-processing steps (note that in the figures, the images are rotated 90 degrees to allow the compact representation of the filtering steps).

I.3.3.1. DCNN DETECTOR

For the DCNN, the training sample was 1000+ of $28 \times 28 \times 3$ greyscale images. The training was achieved using the stochastic optimisation method Adam [14]. The network was trained over 40 epochs with an early stopping patience level of 5, allowing the error to achieve a good minimum. During training, the loss value of angle error was monitored to ensure early stopping whenever the accuracy in the validation set stopped improving. For the entire succession of training, the best model is saved. In training, a dropout rate of 0.25 is implemented to prevent overfitting.

I.3.3.2. GECONV

For the GeConv, the data is fed similarly by applying a random rotation from the initial orientation. In contrast to the DCNN, GeConv did not need any training. Instead, pre-processing filters were chosen carefully, as described in the previous Section. Also, due

to its low computational cost, the geometrical detector could pre-filter and process the images of full size on the fly. The processing time of the pre-processing and the detection was recorded for comparison.

I.3.4. Results

This Section discusses the results of GeConv and the DCNN detectors performed on the image sequences collected from the wind tunnel test. The data set was split into a training/evaluation ratio of 0.8.

I.3.4.1. GeConv

The result of the evaluation of the capabilities of the GeConv is shown in Fig. I.3.5. Here we see the geometric detector's consecutive rotations to convolve the original orientation with the new consecutive rotation angle. Observing the angle from 0-360 with a step size of 5, we see that the symmetry detection for vertical symmetry fires at the point where the distance vectors in the polar plot are aligned vertically. We see that the vertical symmetry is detected at an angle of 97.5 degrees rotation from the initial. This angle can be used to correct the rotation and deduce the symmetry line as we know the global orientation in the frame. Similarly, horizontal detection is fired when the distance vectors are aligned horizontally.

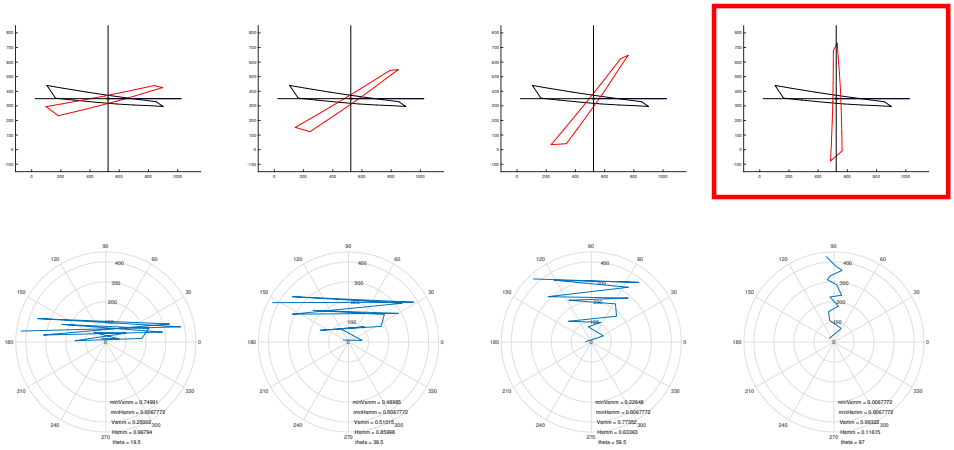


Figure I.3.5: Result of the vertical and horizontal symmetry detection over 0-360 degree rotation from the initial image orientation.

I.3.4.2. DCNN Detector

The training progress is shown over 40 epochs, and batch size 12 is shown in Figs. I.3.6. As seen, the network achieves a near 0.05 angle error over approximately 29 epochs. The early stopping level allows extending the training with the current settings by up to 10 epochs. A dropout rate of 0.25 was implemented for the training due to the relatively small

dataset size. We also see that no discernible overfitting is observable as the validation loss continues to decrease without diverging from training loss. In Fig. I.3.6 it is seen that the loss starts to settle at around 25 epochs.

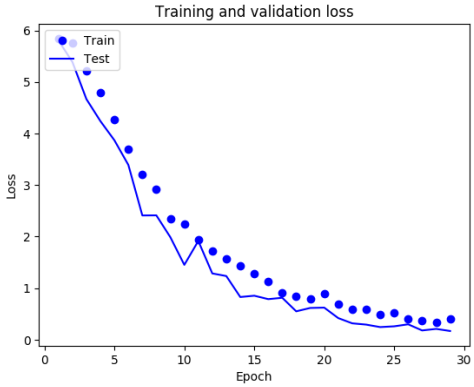


Figure I.3.6: Progress of loss over successive training iterations for training and validation.

The results of the evaluation performed on the 0.2 validation set are shown in Fig. I.3.7. The original orientation is shown; in the middle, the angle is rotated at an arbitrary degree, and the rightmost column shows the predicted angle and the rotation correction. As shown, the DCNN can achieve a perfect prediction of the rotation angle from its initial orientation.

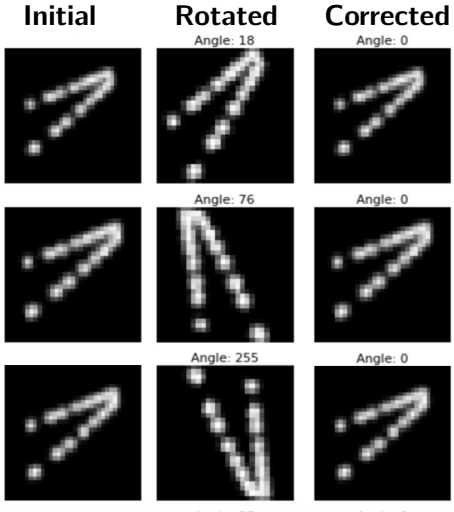


Figure I.3.7: Result of the evaluation of the DCNN over the evaluation data set.

I.3.5. DISCUSSION AND CONCLUSIONS

In this study, a computation of 2-axis reflectional symmetry from the based geometrical relationship of contour points was performed using the proposed GeConv algorithm. The algorithm is tested on a 1000+ image dataset, initialised with rotation, from a wind tunnel and flight test experiment as part of a more extensive study to apply real-time visual tracking of flexible wing structures. Furthermore, an alternative approach is presented using Deep Learning. Here a Deep Convolutional Neural Network is trained to detect the orientation angle of extracted contour shape of the wing. Both approaches show excellent results in detecting symmetry and orientation. The Neural Network can achieve an angle error of up to 0.05 degrees without needing a contour shape as input. However, it does not generalise easily to shapes deviating from the training set. The traditional computer vision approach excels at speed. It is model-free, meaning a good generalisation can apply the algorithm out of the box to many types of shapes. However, it requires a processing step to get the contour points, while the DCNN operates on the raw pixels. Due to its low computational cost, the processing can take more detailed steps to reduce the number of cluster points and filter out irrelevant clusters. The main disadvantage of GeConv is that, while the orientation can be easily corrected, the head or tail of the object along a given symmetry axis is arbitrary. Further tests will be performed to assess the algorithm on arbitrary shapes. One of the possible applications would be lane and runway detection.

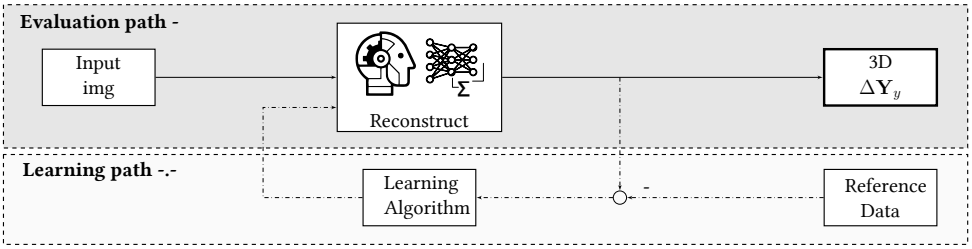


Figure I.3.8: 3D Reconstruction pipeline replaced fully by Deep Learning approach.

Finally, results are shown by the DCNN for symmetry detection of the wing orientation set and step forward towards adopting a deep learning-based approach for the wing shape reconstruction process from raw image data. The adaptability of the DCNN provides the possibility to implement a DCNN processor in other steps of wing shape reconstruction or potentially replace the entire process. This would significantly simplify the processing steps, as shown in Fig I.3.8.

REFERENCES

[1] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Fast Symmetry Detection with Deep Learning and GeConv," in *IEEE RAS 2019 International Summer School on Deep Learning for Robot Vision*, IEEE, 2019.

[2] X. Wu, M. Wand, K. Hildebrandt, P. Kohli, and H.-P. Seidel, "Real-Time Symmetry-Preserving Deformation," in *Computer Graphics Forum*, vol. 33, pp. 229–238, Wiley Online Library, 2014.

[3] Y. Liu, H. Hel-Or, C. S. Kaplan, L. Van Gool, *et al.*, "Computational symmetry in computer vision and computer graphics," *Foundations and Trends® in Computer Graphics and Vision*, vol. 5, no. 1–2, pp. 1–195, 2010.

[4] N. J. Mitra, M. Wand, H. Zhang, D. Cohen-Or, V. Kim, and Q.-X. Huang, "Structure-aware shape processing," in *ACM SIGGRAPH 2014 Courses*, p. 13, ACM, 2014.

[5] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Symmetry descriptors and 3D shape matching," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 115–123, ACM, 2004.

[6] P. Lancelot, J. Sodja, and R. De Breuker, "Investigation of the unsteady flow over a wing under gust excitation," *17th International Forum on Aeroelasticity and Structural Dynamics*, 2017.

[7] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[8] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[9] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Kdd*, vol. 96, pp. 226–231, 1996.

[10] R. Jarvis, "On the identification of the convex hull of a finite set of points in the plane," *Information Processing Letters*, vol. 2, pp. 18–21, mar 1973.

[11] C. Sun and D. Si, "Fast Reflectional Symmetry Detection Using Orientation Histograms," *Real-Time Imaging*, vol. 5, pp. 63–74, Feb. 1999.

[12] F. Chollet, "Keras." https://github.com/fchollet/keras, 2015.

[13] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, 2016.

[14] D. P. Kingma and J. Lei Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations (ICLR)*, arXiv:1412.6980v9, arXiv, 2014.

Appendices

I.A1

Appendices Visual Tracking

I.A1.1. Allegra Wing Specifications

Below the specification of the Allegra wing are presented [1]:

Table I.A1.1: Parameters of the Allegra wing.

Definition	Parameter	Value	Unit
Span	b	1.6	[m]
Top chord	c_{root}	0.36	[m]
Root chord	c_{tip}	0.12	[m]
Mean chord	c_m	0.24	[m]
Taper rati	λ	1/3	[-]
Aspect ratio	$\mathcal{AR}$	6.67	[m]
Sweep (quarter line)	Λ	-17	[deg]
Wing area	S	0.384	[m^2]
Airfoil max. thickness	–	13.028	[%]
Airfoil max. camber	–	2.4	[%]

I.A1

I.A1.2. Tracking Result for Run R3 with Gaussian Noise

Below the results are shown for the tracking sequence from run R3 with injected Gaussian noise (mean of $\mu = 0$ and standard deviation of $\sigma = 0.5$).

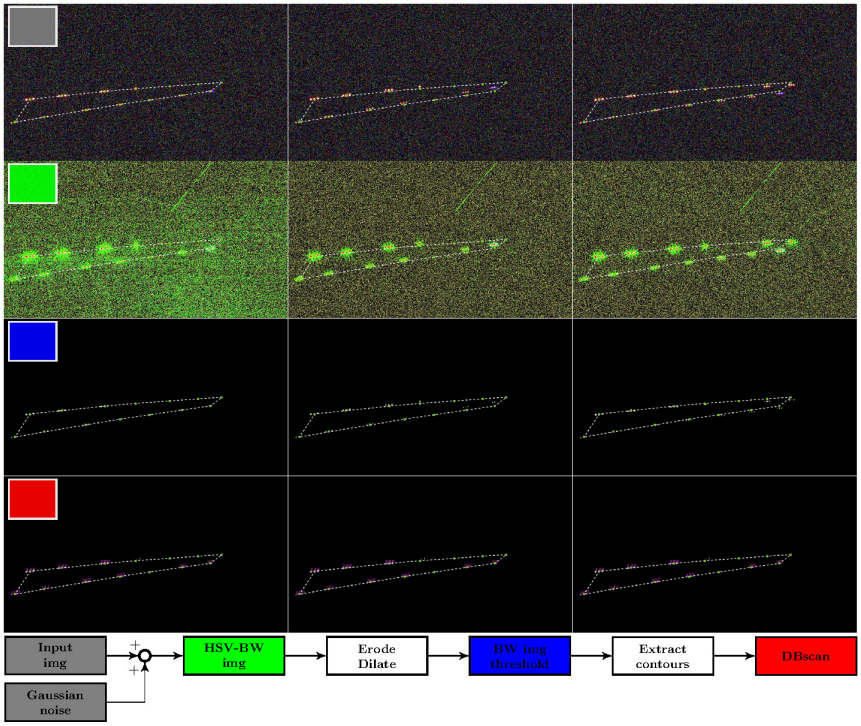


Figure I.A1.1: Tracking sequence on input images (0,60,80) from run R3 with injected Gaussian noise ($\mu = 0$ and $\sigma = 0.5$).

I.A1.3. Deep CNN Network Structure

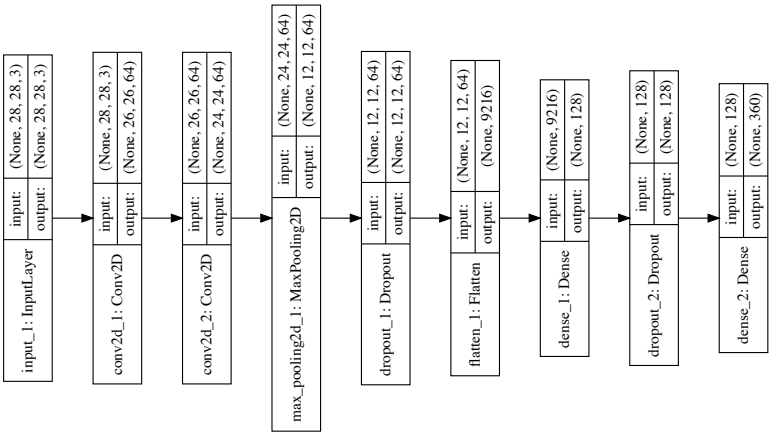


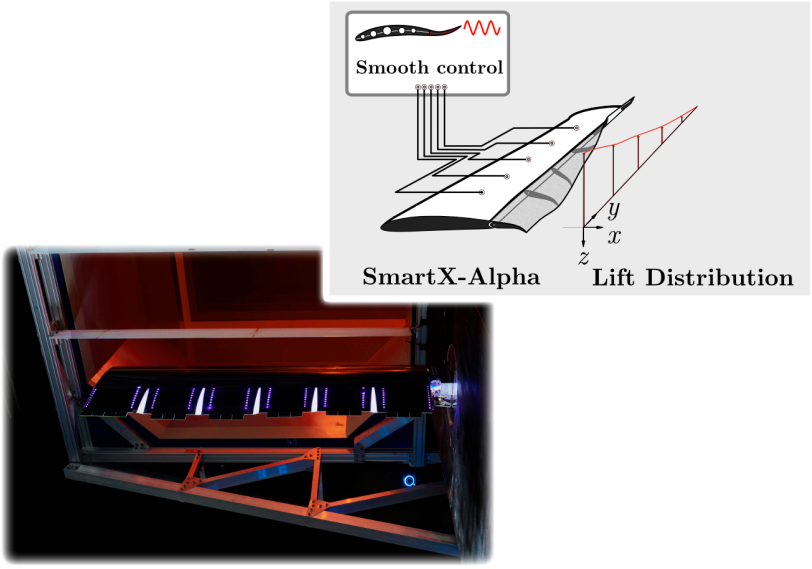
Figure I.A1.2: Network structure of DCNN.

References

[1] Markus Ritter, Yasser M. Meddaikar, and Johannes K.S. Dillinger. Static and dynamic aeroelastic validation of a flexible forward swept composite wing. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2017*, page 0637, 2017.

II

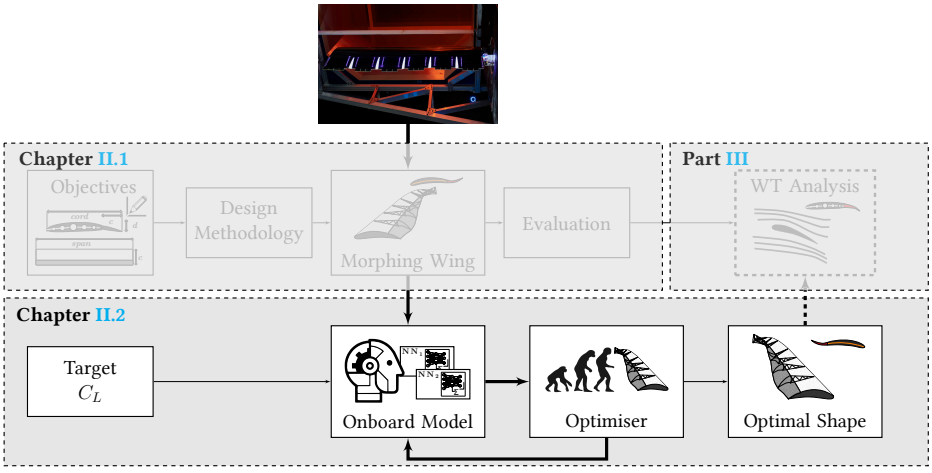
Smart Design & Optimisation



II.1

Distributed Morphing Wing Design

In this Chapter, a novel distributed morphing concept is presented, which addresses the drawbacks of the initial TRIC concept and allows variation of lift distribution locally along the span. The laminate design is optimised with an FSI optimisation framework considering the ply orientation, laminate thickness, laminate properties and actuation loads. Furthermore, a numerical and experimental study is performed to select a suitable elastomer design for inter-modular connection [1].



This Chapter is based on the following peer-reviewed conference and journal paper:
T. Mkhoyan, N. R. Thakrar, R. De Breuker, and J. Sodja, "Design of a Smart Morphing Wing Using Integrated and Distributed Trailing Edge Camber Morphing," in *ASME 2020 Conference on Smart Materials, Adaptive Structures and Intelligent Systems; In Review at Smart Materials and Structures Journal*, ASME, sep 2020

II.1

This Chapter investigates the design and development of an autonomous morphing wing concept developed within the scope of the SmartX project, which aims to demonstrate in-flight performance optimisation with active morphing. This chapter proposes a novel distributed morphing concept with six translation-induced camber morphing trailing edge modules to progress this goal. The modules are interconnected using elastomeric skin segments to allow seamless variation of local lift distribution along the wingspan. A fluid-structure interaction optimisation tool is developed to produce an optimised laminate design considering the ply orientation, laminate thickness, laminate properties and actuation loads of the module. Analysis of the kinematic model of the integrated actuator system is performed, and a design is achieved which meets the required continuous load and fulfils both static and dynamic requirements in terms of bandwidth and peak actuator torque with conventional actuators. The morphing design is validated using digital image correlation measurements of the morphing modules. Characterisation of mechanical losses in the actuator mechanism is performed. Out-of-plane deformations in the bottom skin and added elastomer stiffness are identified as the impacting factors of the reduced tip deflection.

Nomenclature

Acronyms

FEM	=	Finite Element Model
FSI	=	Fluid-Structure Interaction
TE	=	Trailing Edge
LE	=	Leading Edge
TRIC	=	Trailing Edge Induced Camber
FBD	=	Free-Body Diagram
DIC	=	Digital Image Correlation
AOA	=	Angle of Attack, $^\circ$

Symbols

c_p	=	aerodynamic pressure coefficient
C_L	=	coefficient of Lift
F_a	=	actuator force, N
F_l	=	linkage force, N
F_r	=	reaction force at linkage, N
r_a	=	torque arm length, m
δ_a	=	actuator rotation, $^\circ$
δ_l	=	angle between F_a and F_l, $^\circ$
T_a	=	actuator torque, [Nm]
x_a	=	actuator horizontal deflection range, Nm
x,y,z	=	nodal displacement along x,y,z axes, m
V_∞	=	air speed, m/s
rho	=	air density, kg/m^3
M_x, M_y, M_z	=	nodal moment around x,y,z axes, Nm

Subscripts

1,2	=	actuators left and right of module
lin	=	linear coefficient
non-lin	=	non-linear coefficient
lin,non-lin	=	linear, non-linear coefficient
max,min	=	maximum and minimum values

Superscripts

aero	=	coefficient calculated on the aerodynamic mesh
fem	=	coefficient interpolated on the FEM mesh
max,min	=	maximum and minimum values

II.1.1. Introduction

The advancements in aerospace materials, manufacturing technology, controller and hardware design allow for developing increasingly lighter and more complex concepts such as morphing wings, which significantly benefit flight performance. Initially inspired by avian biology, the effectiveness of such concepts is demonstrated in nature, with wing shape adaptation and optimal gliding performance [2, 3].

As in nature, morphing wing concepts have evolved since the early years of aviation. One of the well-documented examples was the active roll control of the Wright Flyer, the first successful heavier-than-air powered aircraft. In this lightweight design, the lateral stability was ensured by wing twist-warping [4]. In this case, the flexible fabric-wrapped structure was well suited for morphing. However, as the flight speeds and loads increased with the advancement of aircraft design, a stiffer wing was required to fulfil load requirements and overcome aeroelastic instabilities. As a result, the conventional rigid wing design, generally optimised for cruise, exhibits compromised performance in other flight conditions. More importantly, due to continuous fuel burn and redistribution of the weight, no optimal configuration can be found which is met through the entire cruise phase.

Active morphing has the potential to reduce this performance gap and continuously optimise the aircraft performance across the entire flight envelope adaptively. However, a challenging aspect in active morphing is designing a feasible and effective morphing mechanism such that the aircraft performance can be improved actively throughout the flight envelope [5]. That is the subject of this Chapter.

In literature, various morphing concepts can be found. A comprehensive review of the early morphing concepts of various approaches regarding the actuator material, the actuation mechanism, and the skin types is summarised by Barbarino et al. [6]. Examples vary from conventional to compliant mechanisms and materials in the latter two categories. Also, various materials are investigated for the actuators, ranging from conventional to piezoelectric and shape memory alloys. In aircraft wings, morphing can be applied to the leading edge, trailing edge, or both.

Kintscher et al. and Sodja et al. investigate a seamless morphing droop nose concept for the leading edge, designed to match a given target shape with different materials used for the morphing skin such as glass-fibre pre-preg and aluminium [7, 8]. The concept by Sodja et al. utilises conventional actuation. Here, low actuation forces are achieved by maintaining the skin length constant during morphing, such that strains in the skin are kept minimal. Several other concepts achieve low actuation force by utilising compliant skin and actuation mechanisms [9–11]. While promising, the studies highlight the importance of further research into manufacturing and up-scaling complex compliant designs since the manufacturing process of these complex shapes is still challenging.

Further, examples of the compliant mechanism and actuation are investigated. Previtali et al. used conventional actuators, and Molinari et al. used piezoelectric skin actua-

tion [10–13]. Some studies use bio-inspired design, such as the FishBAC concept, designed to mimic the compliant skeletal frame of fish, developed at Swansea University [14–16]. Trailing edge mechanisms are also presented by FlexSys, which have been installed and undergone flight tests on a modified Gulfstream III business jet [10, 17, 18].

Recent studies also investigated the use of ultralight, lattice-based structural modules assembled in a modular adaptive structure using carbon fibre-reinforced polymers (CFRP) [19–21]. The advantage is that these materials can have the stiffness of a typical elastomer at the mass-density typical to aero-gel. Cremer et al. [20] demonstrate improved aerodynamic efficiency and roll control authority with spatially programmed elastic morphing shape of a $4.27\ m$ wingspan aircraft in the wind tunnel. Jennet et al. [19] present the digital morphing wing concept constructed from discrete lattice elements. This concept shows increased roll efficiency compared to a conventional wing by applying spanwise twisting deformation. While promising, due to its programmable flexibility and lightweight, the lattice-based modules occupy most of the internal space. Therefore, additional consideration is needed to ensure the flexibility of the structure while reserving the room for fuel, batteries, and other components to be installed in the wing. The lattice-based concept, presented by Keidel et al. [21], suggests a potential structurally efficient approach through optimisation of the orientation and distribution of the CFRP rods. However, additional consideration is needed for larger wing structures and manufacturability aspects for this concept.

In addition to the internal actuation mechanisms, the concept of morphing through direct skin actuation was investigated by Bilgen et al. with piezoelectric actuators [22, 23]. Pankonien et al. investigated skin actuation with macro fibre composite actuators for camber morphing [24]. Another wing concept by Mistry et al. demonstrated a cross-sectional warping mechanism to realise variable camber on a rotor blade [25].

Another study developed under NASA Advanced Air Transport Technology investigated the multi-flap Variable Camber Continuous Trailing Edge Flap (VCCTEF) concept for the Generic Transport Model (GTM) [26]. This concept demonstrated effectiveness in multi-objective control and gust load alleviation in studies [27, 28]. However, the real-life experimental demonstrator with SMA rotary actuators revealed many challenges such as weight effectiveness, speed and power requirements of the actuators, the complexity of the multi-segment camber mechanism, and skin flexibility required [26].

In a recent study conducted in the EU FP7 CHANGE project, a morphing concept called the Translation Induced Camber (TRIC) is introduced to address some of these problems [29]. This concept implements a relatively simple and effective morphing mechanism that uses a combination of cross-sectional warping and skin bending to induce both camber and twist morphing with a pair of conventional actuators. The advantage of this concept is its relative simplicity and compactness of the actuation mechanism, which increases the fuel carrying capability and volume needed for necessary auxiliary components in the wing. However, the main disadvantage of the current TRIC concept is that the lift distribution cannot be influenced locally with a single morphing surface controlled by a single pair of actuators. As a result, this inhibits the use of the morphing mechanism for multi-objective flight control and limits its use as a direct replacement of conventional control surfaces for rigid body motion control (ailerons, rudders and elevators). Various control design studies highlight the necessity and effectiveness of multi-objective flight control,

load alleviation, and drag reduction performed by distributed multi-flap systems such as the VCCTEF in [27, 28] and conventionally flapped over-actuated aircraft models [30].

In summary, the literature survey suggests that many morphing concepts are restricted to either wing twist or camber morphing mechanisms, proposing a complex mechanism that introduces manufacturing challenges or consumes a large portion of the wing's internal volume. Furthermore, most concepts show a global morphing approach, while in the scope of active control, a distributed and over-actuated mechanism is necessary to apply simultaneous gust and manoeuvre load alleviation, flutter suppression and drag minimisation.

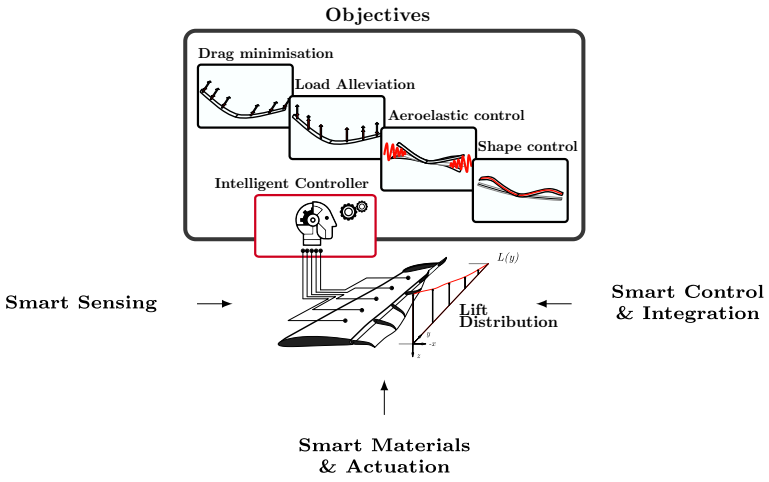


Figure II.1.1: Rationale of the SmartX wing [1].

The current study extends the TRIC concept to address this problem. It introduces a distributed and modular morphing design interconnected with elastomeric skin segments to allow seamless active twist and camber morphing. This way, the lift distribution along the wingspan can be varied locally and actively without additional drag penalty due to the gaps between the control surfaces. Furthermore, it allows deploying multiple control surfaces for various control objectives. This morphing design was developed in the scope of an autonomous smart wing project called SmartX, which aims to demonstrate an integrated and coherent approach to multi-objective load alleviation, flutter suppression and performance optimisation of adaptive aircraft wings.

In Fig. II.1.1 the rationale behind the integrated design of the SmartX wing is presented [1]. The purpose of this technology demonstrator is to demonstrate performance optimisation of multiple objectives such as (i) drag optimisation, (ii) load alleviation, (iii) flutter suppression and (iv) shape control through multidisciplinary utilisation of smart sensing, control, actuation, and integration [31].

Addressing the shortcomings of the previous morphing designs, the contributions of this Chapter are threefold. First, a distributed morphing concept is developed, analysed, and validated using Digital Image Correlation (DIC) measurements. Second, solutions are investigated and implemented to improve the aerodynamic character and the conti-

nuity between adjacent morphing modules. Finally, a computationally efficient design framework is developed for analysis and design optimisation. This Chapter aligns these objectives in the scope of the development of the SmartX-Alpha wing demonstrator.

This Chapter is structured in the following way. First, the scope of the SmartX project is presented in Sec. II.1.1, followed by a brief overview of the TRIC and the design evolution of the distributed morphing concept in Sec. II.1.2. The design methodology describing the development of the Fluid-Structure Interaction (FSI) tool and the design optimisation framework are presented in Sec. II.1.3. The demonstrator manufacturing and integration are presented in Sec. II.1.4. The finalised design and the design validation are shown and discussed in Sec. II.1.5. Finally, conclusions are drawn in Sec. II.1.6.

II.1.2. Morphing Concept

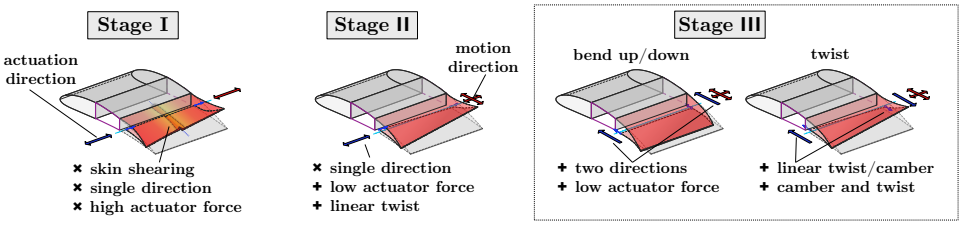


Figure II.1.2: TRIC morphing concept design evolution stages I-III by Werter et al. [29].

The TRIC concept served as the basis of the design [29]. The morphing concept underwent several stages of evolution, which led to the development of a fully composite smart morphing wing concept, named the SmartX-Alpha demonstrator. The motivation for the development of this demonstrator is explained in Sec. II.1.1, the design evolution of distributed TRIC concept is discussed in Sections II.1.2.1 and II.1.2.2.

II.1.2.1. TRIC Overview

The literature review highlighted some shortcomings of previous morphing designs, which are addressed to a degree by the TRIC concept developed in CHANGE [29]. The crux of the TRIC is to utilise a combination of cross-sectional warping and skin bending to induce camber and twist morphing powered by conventional actuators. Due to its relative simplicity and compactness, the mechanism is economical in weight and size. The following section explains how the TRIC concept has evolved from design stages I-III, as illustrated in Fig. II.1.2. In this figure, sections of the morphing wing are depicted, with the red area indicating the morphing skin and the purple outline, the wing box. The direction of actuation is indicated with blue arrows, and the direction of motion of the skin in red.

TRIC concept is derived from the principle of a warping cross-section [32], first introduced by Vos et al. with a twist morphing concept. The first design stage of TRIC, similar to the latter concept, introduces a cut along the span on the bottom of the wing and utilises spanwise actuation without any chordwise movement (Stage I of Fig. II.1.2). The spanwise actuation provides the intended warping of the skin; however, since the wingbox is relatively stiff in comparison, shearing of the skin is also introduced. This re-

sults in relatively large actuation forces and non-linearity in spanwise twist distribution. In the next stage of design evolution, Stage II in the middle of Fig. II.1.2, chordwise motion is allowed, as indicated by the red arrow. As a result, shear deformations are significantly reduced, leading to lower actuation forces and smooth linear twist distribution along the span. However, only twist morphing is possible since the chordwise motion is coupled to spanwise actuation. In the final stage, stage III, the actuation direction is changed from spanwise to chordwise, and both camber and twist morphing can be commanded independently while the actuation forces are low. By altering the actuation direction, two sets of actuators moving either symmetrically (in the same direction) or asymmetrically (in the opposite direction) can now introduce pure camber morphing (Bend Up/Down) or warp-induced spanwise twist morphing (Twist), as shown in the last column of Fig. II.1.2.

II.1.2.2. Distributed Morphing

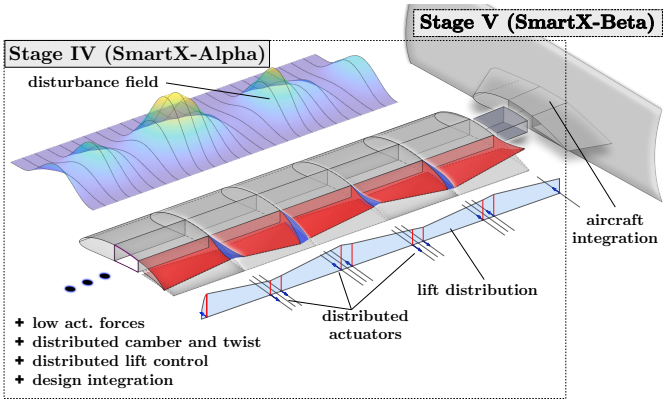


Figure II.1.3: Distributed TRIC morphing concept, the SmartX-Alpha, investigated in this Chapter.

In stage III, the TRIC concept achieves an efficient compact design, where the actuation forces are kept low, and the internal space is not compromised. While promising, a single-pair TRIC actuator design has a significant limitation: a single control surface spans the entire wing. Hence, neither camber nor twist can be controlled locally.

Therefore, from the perspective of control design, the actuator system cannot satisfy multiple objectives simultaneously. Generally, the typical approach is to separate the control tasks over the available control surfaces when the control design must address multiple objectives [33, 34]. To prevent conflicts between various tasks such as pitch control, roll control or load alleviation, the control surfaces are either assigned to specific tasks [35]. Another possibility for conventionally actuated flexible aircraft is to make a compromise between the objectives, as reported in [36]. The conflict can be prevented if the aircraft features a sufficiently high number of control surfaces such that several control tasks can be addressed simultaneously. This would allow addressing both the attitude control of the aircraft body (pitch, roll, yaw) and aeroelastic control (load alleviation, flutter control etc.) by continuously adjusting the same set of control surfaces (control allocation) [30]. When combined with morphing, more objectives can be achieved in control architecture, such

as shape and drag optimisation, leading to a more optimal lift distribution. Variable Camber Continuous Trailing Edge Flap (VCCTEF) is an example of a multi-segment camber morphing concept integrated on a flexible wing Generic Transport Model (GTM), where drag minimisation is demonstrated [27, 28]. However, the complexity of this multi-hinged distributed morphing design has restricted the development of the VCCTEF to mainly numerical studies.

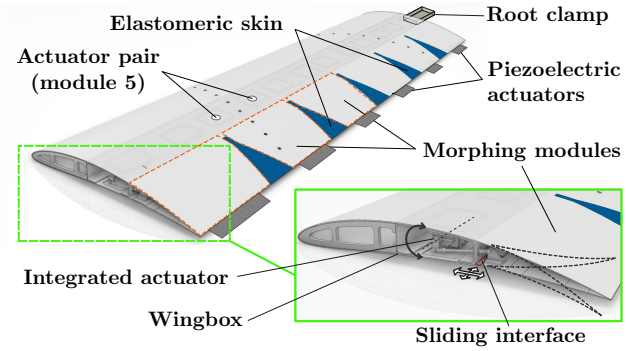


Figure II.1.4: Overview of the SmartX-Alpha seamless TRIC morphing concept.

Our study aims to contribute to the state-of-the-art morphing design and proposes a distributed modular morphing wing concept suitable for multi-objective control. The proposed concept extends on the simplicity of the TRIC concept and benefits from the smooth over-actuated morphing system. In this context, it investigates a solution for the discontinuity introduced by a distributed multi-flap system. In addition, the aim is to demonstrate the concept of a wind tunnel model called the SmartX-Alpha. This demonstrator includes six morphing modules, allowing the independent camber and spanwise twist morphing of local span segments. This design evolution is illustrated in stage IV in Fig. II.1.3.

As with the TRIC concept, the skin is actuated internally, allowing smooth and seamless morphing along the chord. The morphing target shape is commanded using fast, high torque servos embedded in the wing box. The servos allow the trailing edge bottom skin to slide chordwise and spanwise inside a guiding slot, as illustrated in the green zoomed-in box in Fig. II.1.4. Each module has two actuators, allowing local symmetric (pure bending) and asymmetric (twist) morphing. Interconnected triangular skin segments, joined by an elastomer material, allow for continuous spanwise variation of the morphing wing shape, facilitating a continuous lift distribution over the wingspan. The triangular skin segments, illustrated in Fig. II.1.4, are composed of elastomeric material.

The main advantage of the current design is that by adjusting the camber and twist distribution, the lift distribution can be controlled locally and independently for each module, allowing the wing to assume an optimal aerodynamic configuration to maximise the lift-to-drag ratio to minimise drag. Furthermore, the lift distribution can be adapted to perform manoeuvre load alleviation when necessary by redistribution of the lift closer to the root of the wing. Lastly, fast piezoelectric actuators are placed at the tip of the morphing trailing edge for aeroelastic control such as flutter suppression, hereby covering the targets presented earlier in Fig. II.1.1. To sustain the required loads, meet the actuator constraints,

and achieve the desired morphing shapes, the composite skin of the wing is tailored and optimised. Therefore, a Fluid-Structure Interaction (FSI) structural optimisation tool is developed to allow fast analysis and optimisation of ply orientation and laminate thickness in terms of the given input loads, desired target shapes and actuator limits. The devised design methodology used in this assessment is presented in the next section.

II.1.3. Design Methodology

The design methodology is centred around the FSI tool, which connects the Finite Element Model (FEM) and the Aerodynamic model (XFOIL). It has the following goals in mind: (i) to produce a detailed morphing design presented in Sec. II.1.2.2 and (ii) to allow quick assessment of the wing performance under given aerodynamic load and commanded actuators input position for data-driven surrogate model and control design.

II.1.3.1. FSI Framework

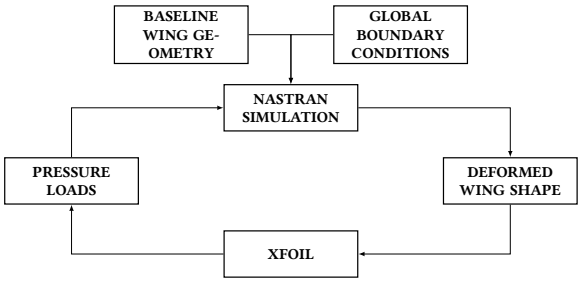


Figure II.1.5: Flow chart of the FSI framework [1].

The general flow diagram of the FSI framework is illustrated in Fig. II.1.5 [1]. The NASTRAN FEM model [37] and the aerodynamic model represented by XFOIL [38] are coupled by the framework, in which MATLAB acts as the interfacing software.

The actuator displacement was used as input to the simulation for each loop iteration. The loop was iterated until the deformed shape, lift, and actuation loads converged. As shown in Fig. II.1.6 convergence of these variables is typically reached in three to four iterations [1]. Here the progressive variation of these variables is shown over each iteration. The dotted blue line indicates the deformation convergence criterion of $1e-3$, evaluated as the cumulative variance of deformations over each node (continuous blue line). The convergence criteria and the convergence study are further discussed in Sec. II.1.5.1.

The actuator displacement was modelled by displacing the nodes, which acted as the interface between the actuator's pushrod and the skin. This is depicted in Fig. II.1.7. The actuator displacement would, in turn, impose a deformation of the shape, which was extracted from NASTRAN and transferred to XFOIL to exact pressure distribution data. The pressure data was returned to the NASTRAN model as static pressure loads.

FEM

The FEM model, representing a morphing module of 500 mm chord and 300 mm span, was developed in NASTRAN [37]. A cut is introduced at the bottom skin, and nodes are

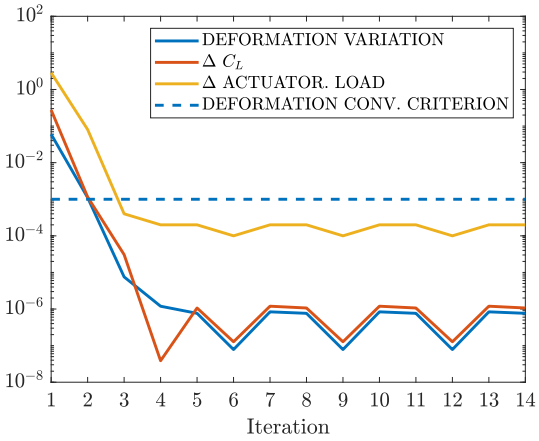


Figure II.1.6: FSI framework chart and convergence [1].

selected to introduce the actuation loads, according to the TRIC principle. To model the morphing principle, the flexible skin is allowed to slide in the chordwise and transverse direction to accommodate the required morphing shape under loads. Vertical deformations and rotations about the transverse and chordwise axis are restrained. Fig. II.1.7 shows the boundary conditions employed in the FEM model.

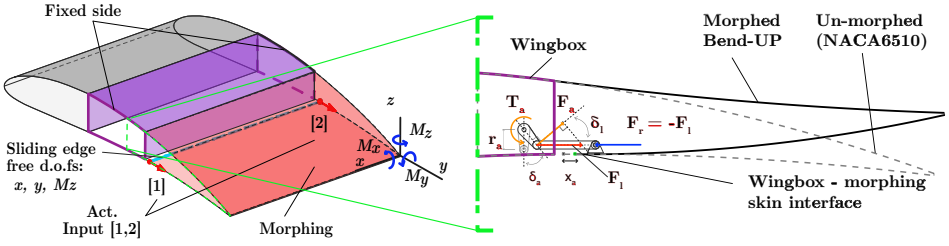


Figure II.1.7: FEM model boundary conditions and actuator free body diagram.

These boundary conditions are chosen such as to allow the following morphing states, which are referred to as morphing subcases in the FSI model: (i) Bend Up, (ii) Bend Down, (iii) Twist.

Actuation Mechanism

A kinematic model of the actuation mechanism and the morphing interface has been constructed to facilitate the actuator sizing and the mechanical behaviour. The model presented in Fig. II.1.7 shows (in a cross-sectional view) the Free-Body Diagram (FBD) of the actuator force and moment equilibrium for the Bend Up case. The input to the actuator is the commanded angle, which in turn results in specific actuation torque (required to reach the desired position). Two essential aspects can be noted regarding the actuator input force to the morphing interface when examining the diagram: (i) the maximum ac-

tuator force, F^{max}, that can be delivered at the attachment on the servo arm will increase by shortening the servo arm r_a, (ii) the instantaneous actuator torque T_a required to balance or overcome the reaction forces at the morphing interface, F_r, is not linear for the range of travel of the actuator. The latter aspect results from the kinematic relationship of F_a, representing the actuator torque for a given actuator arm r_a and the linkage force F_l. Since F_a is a projection of the F_l normal to the servo arm ($F_a < F_l$), F_a and thus the amount of torque required by the actuator to balance or overcome the reaction forces are dependent to the relative position of the linkage system concerning the servo arm.

The relation of linkage force to δ_a is presented in the side view in Fig. II.1.7, and is defined by the following relation:

$$F_l = \frac{F_a}{\cos(\delta_l)} \approx \frac{F_a}{\cos(\delta_a)} \tag{II.1.1}$$

As the linkage length will be much greater than the arm's length, δ_l can be approximated by δ_a.

Following the expression above, the actuator torque can be calculated as:

$$T_a = F_l \cos(\delta_l)\, r_a \tag{II.1.2}$$

In this expression, the r_a represents the length of the torque arm, and the non-linear relationship between the actuator torque, T_a and the rotation angle, δ_l is reflected here by the cosine term. The following step was to establish an optimal torque arm such that the actuation loads are kept minimal within the desired morphing shapes, which is explained in Sec. II.1.5.3. The shapes are governed by the horizontal deflection range of the actuator (along F_l), indicated as x_a in Fig. II.1.7.

II.1.3.2. Aerodynamic Model

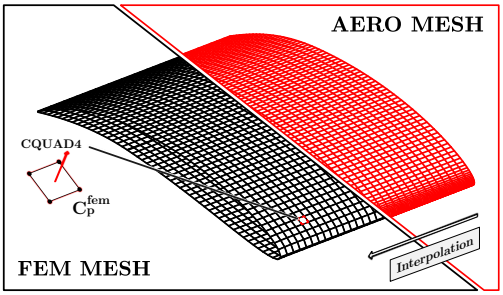


Figure II.1.8: Mesh comparison FEM (left) and aero (right).

Aerodynamic behaviour is modelled using a 2D airfoil analysis based on strip theory. At each FSI loop iteration, the deformed wing shape is sliced into discretised spanwise sections to extract 2D deformed airfoils. The shape of the deformed airfoil is the input to XFOIL, which calculates the pressure distribution over the deformed airfoil [38]. This data is then used to calculate and interpolate the pressure loads corresponding to the structural

II.1

mesh through the slicing of the airfoil. In NASTRAN, aerodynamic force is applied as static pressure load (PLOAD), defined as a uniform pressure load on the quadrilateral surface (CQUAD4) comprised of four nodal positions, indicated in the bottom part of Fig. II.1.8. Slice planes coincide with NASTRAN grid points to limit interpolation routines. This process of airfoil extraction is shown on the right of Fig. II.1.9. The aerodynamic mesh is

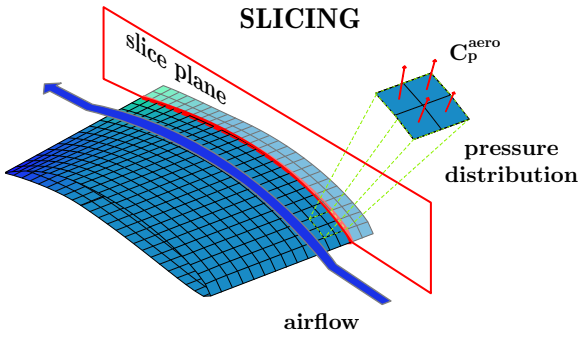


Figure II.1.9: Airfoil slicing process.

denser than the FEM mesh to improve the resolution of the pressure distribution, particularly in the leading and trailing edge area. An additional interpolation routine is deployed to deal with the differences in meshes and sample the pressure loads on the corresponding quadrilateral surfaces of the structural mesh. This process is explained in the next section; the difference in the structural and aerodynamic mesh is presented in Fig. II.1.8.

System Coupling

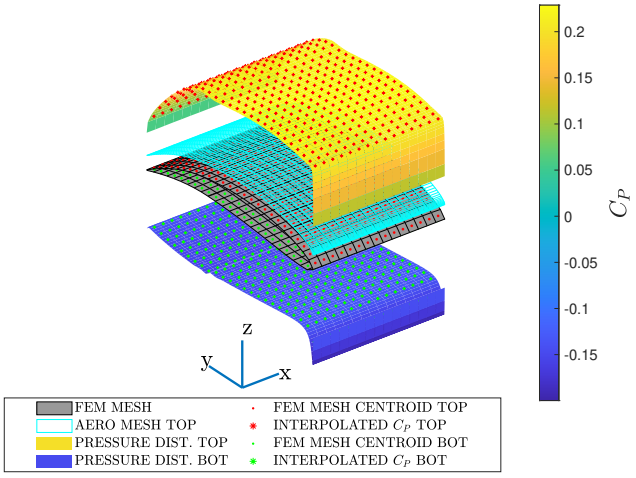


Figure II.1.10: Coupling of data between aerodynamic and FEM models [1].

As the meshes vary between the two models (i.e. the aerodynamic model has a higher resolution), an interpolation routine was used to transfer the pressure coefficient of aero-

dynamic mesh c_p^{aero}, to their appropriate centroid location in the FEM mesh c_p^{fem} values (Fig. II.1.8). The interpolation function implements a gridded linear interpolation method based on Delaunay triangulation of the 2D grid data [39]. Since the interpolation is performed in 2D, the airfoil's upper and lower pressure distributions are sampled from the aerodynamic mesh separately and projected independently onto the x, y projection of the FEM mesh, acting as xy data sampling points for the interpolation. These sampling locations are the centroid of the mesh elements, as is illustrated in Fig. II.1.10 [1].

The interpolation method uses MATLAB's griddata function. The interpolated pressure coefficients are then converted to pressure loads by evaluating the area of the corresponding mesh element. This is achieved using the NASTRAN PLOAD card [40], which allows evaluating static pressure load directly from pressure coefficients. Appropriate scaling must be applied with the airspeed for the aerodynamic analysis. The conversion from non-dimensional pressure coefficient, cp to PLOAD pressure is performed as follows:

$$P_{LOAD} = -c_p^{fem} \frac{1}{2} \rho V_\infty^2 \tag{II.1.3}$$

The result of the interpolation routine can be visualised in Fig. II.1.10.

II.1.3.3. Design Optimisation

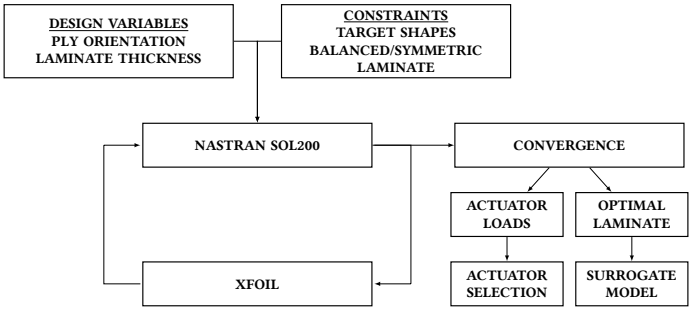


Figure II.1.11: FSI optimisation framework [1].

An optimised laminate design was desired, capable of attaining the three target morphing conditions utilising a minimum actuation load. For this purpose, an optimisation framework was constructed, which is presented in Fig. II.1.11 [1].

Optimisation Parameters

The core of the optimisation framework consists of NASTRAN's SOL200 optimiser [40], implementing the same FSI analysis strategy as shown in Fig. II.1.5. For conciseness, the additional input blocks to the FSI loop are left out; the aerodynamic interaction is facilitated by XFOIL using the same convergence criteria. The three morphing conditions were considered subcases in the optimisation routine and were simultaneously optimised.

Overview input parameters, design variables and constraints used in the optimisation are presented in Tab. II.1.2. Here the ranges of input variables correspond to each subcase, carried out at V_∞= 35 m/s. For Bend Down and Up cases, upper and lower limits of the

Table II.1.2: Overview of input parameters, design variables and constraints of the optimisation.

	variable	range	unit
input	angle of attack (AoA)	[-12,0,5]	[°]
	deformations	[30,-20,±20]	[mm]
	V_∞	35	[m/s]
design	ply angle	[-90,-45,45,90]	[°]
	thickness	[1.65,0.15]	[mm]
range	laminate properties	balanced/symmetric	[-]
	target shapes	[Bend Up/Down, Twist]	[–]

angle of attack (-12°,5°) were used to consider the most adverse aerodynamic loads within the stall limits of the airfoil. The range of deformations was accordingly to maximise the lift coefficient change, ΔC_L, from the undeformed NACA6510 state, within these stall limits. An overview of lift coefficients for various subcases is presented in Tab. II.1.5.

The optimisation outputs were the actuator loads needed for sizing the morphing mechanism and laminate design defined by ply orientation, layers and thicknesses.

Optimisation Model

Several optimisation strategies were investigated, which varied either the ply angle, the laminate thickness, or both.

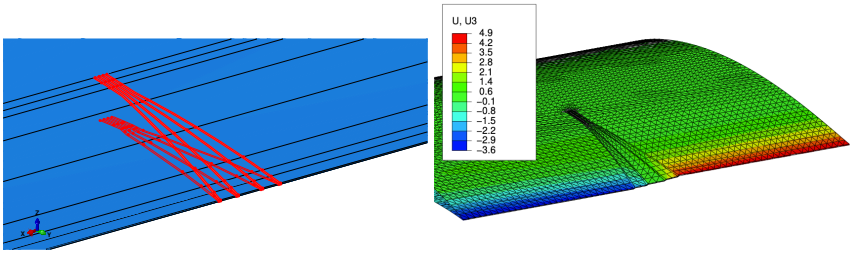
Optimising both the ply orientation and the thickness across discretised strips of the morphing surface proved to be the least successful strategy for two reasons. Firstly, the high number of variables made it reasonably difficult to converge to a solution. Secondly, the ply continuity across the strips could not be ensured without additional constraints, which produced an unrealistic design for hand layup. When only the ply orientation was varied against a constant thickness, the resulting curvature of the morphed shapes exhibited kinks along the cord.

The simplest approach, where the ply thickness gradually dropped with the ply orientation fixed at ±45, produced good results; gradually varying the thickness allowed sufficient flexibility chordwise to eliminate the kinks. Orienting the fibres along the span minimises the actuation force; however, this also compromises the chordwise strength. The ±45 ply orientation provided the best compromise between the actuation force, bending stiffness and torsional stiffness. Furthermore, this configuration allowed symmetrically balanced ply drops and was easily manufacturable from readily available woven plies. Several laminate properties were investigated, and fibre-glass designation US 7630 (MIL-Y-1140H) presented satisfactory results and was selected for use in the final design [41]. This finalised design is discussed in Sec. II.1.5.3.

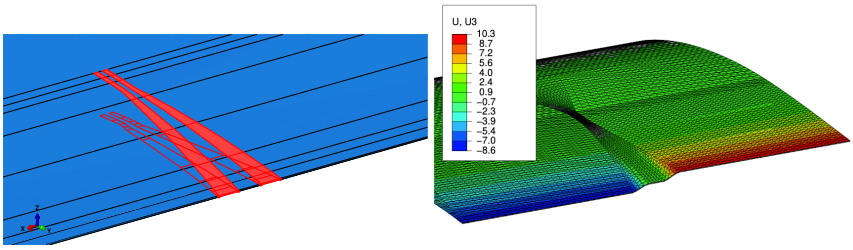
II.1.3.4. Elastomeric Skin Design

A flexible connecting skin segment was investigated to prevent gaps between the morphing modules and ensure independent actuation between adjacent modules. The requirements for the connecting segments were to (i) ensure continuous morphing while not exceeding the continuous load requirements of the selected actuators, (ii) ability to sustain pressure loads for the operational flow regimes (up to 50 m/s), (iii) allow post-

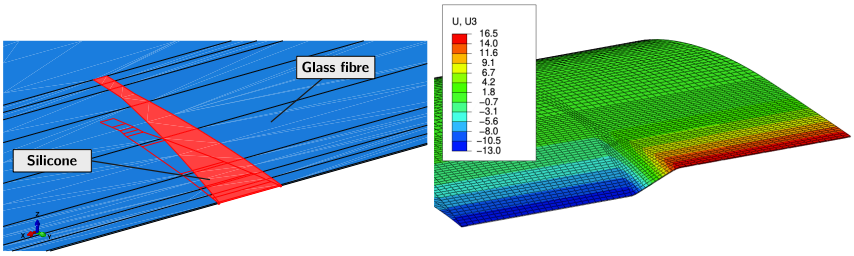
manufacturing integration and (iv) repair. Silicone material was selected for the skin as a connecting body due to ease of use and a good compromise between strength and durability [42].



(a) Narrow slits filled with silicone.



(b) Wide silicone patch reduced slit density.



(c) Gap fully covered by silicone.

Figure II.1.12: Elastomer skin design concepts.

FE models were built in Abaqus to study the elastomer material's flexibility and impact on the actuation loads. Various skin patch designs were considered with silicone-filled slits of varying width and slit density, as shown in Fig. II.1.12. Through assessment of the morphing configurations, it was established that the critical aspect in elastomer integration was the amount of skin shearing required for asymmetric actuation between adjacent modules. The analysis with the FEM model revealed that the best compromise was a complete gap filled with an elastomer material. Further assessment was done using sample testing, as explained in Sec. II.1.4.2.

II.1

II.1.4. Design Integration and Testing

The wing design discussed in Sec. II.1.3 was used to build the wing demonstrator. A manufacturing procedure was established to allow the simultaneous integration of sensors and the optimised design implementation. This process is schematised in Fig. II.1.13.

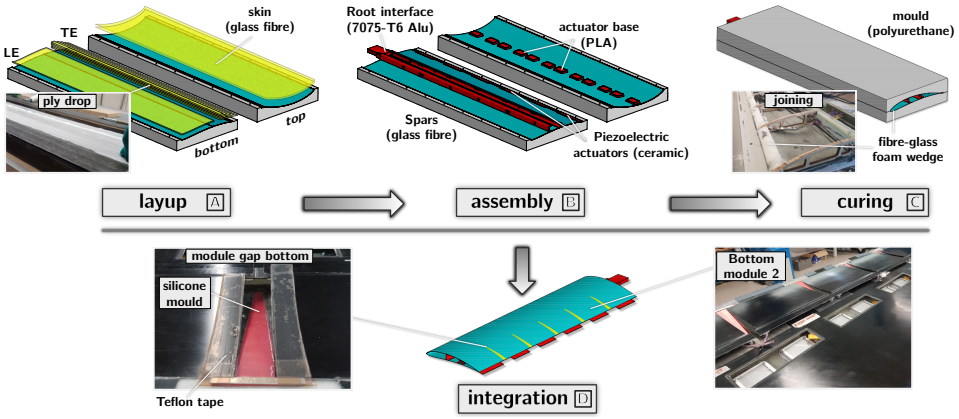


Figure II.1.13: Wing manufacturing and integration process.

II.1.4.1. Manufacturing Approach

The wing design comprised four parts (i) top skin, (ii) bottom skin, (iii) wing box and spars, (iv) morphing trailing edge. The wing skin was manufactured by vacuum-curing wet-laid glass fibre inside top and bottom polyurethane moulds. This manufacturing and assembly process is shown in sub-figures A-D of Fig. II.1.13. The top skin was made in one pass, with the ply dropping incorporated at the trailing edge (Fig. II.1.13-A). Due to the cut incorporated for the TRIC sliding edge, the bottom skin was made in two curing steps, the wing box skin and the morphing trailing edge.

Figures II.1.13-B, II.1.13-C and II.1.14 show the joining process and the assembly of various components in two wing halves. Figure II.1.13-D shows the final integration process. The skin seam along the trailing and leading edge were additionally reinforced with a fibre-glass wrapped foam wedge.

II.1.4.2. Elastomeric Skin Assessment

During the assessment of various silicone skin configurations, it was established that high-density slits, as shown in Fig. II.1.12a, did accommodate sufficient skin shearing to allow opposite actuation of adjacent modules within desired limits. It was observed that gradually increasing the gap between the adjacent modules and decreasing the slit density allowed more skin shearing, as suggested in Sec. II.1.3.4. Further, prototyping and testing using a 3D-printed jig validated the final elastomer design with a fully silicone-filled gap leading to the best results (Fig. II.1.12c). The prototyping jig shown in Fig. II.1.15 was actuated with two Volz DA-22-12-4112 servos and represented the connection between the adjacent modules.

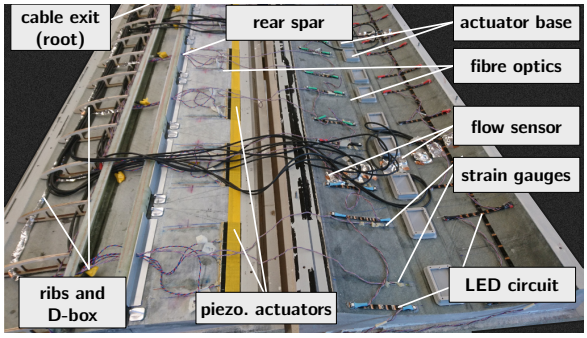


Figure II.1.14: Assembly and integration of smart sensors, actuators and components in the wing.

The jig was used to test the maximum deflection in Bend Up/Down and Twist morphing of various silicone skin samples from the Wacker Elastosil series. The moisture-curing rubber silicone Wacker E41 [43] provided the best compromise between flexibility and bonding durability.

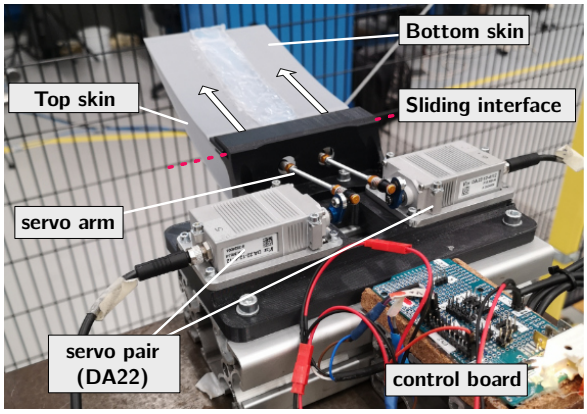


Figure II.1.15: Testing jig components.

Once the silicone material and configuration were finalised from the jig, a manufacturing procedure was established to reproduce the design. A combination of 3D printed PLA (Polylactic Acid), and Teflon tape was used as a mould and release agent, respectively, defining the curvature between the module gaps. In the left and right of Figure II.1.13-D, the application process of the silicone is shown with the modules facing downward and the silicone applied from inside.

After curing the silicone and the final assessment of the elastomeric skin on the wing demonstrator, it was established that the joint was stiffer in shear than anticipated from the numerical analysis explained in Sec. II.1.3.4. A feasible limit for the actuation was between $\pm 25°$. The manufactured demonstrator was subject to morphing characterisation tests explained, and further discussion is made in Sec. II.1.5.4.

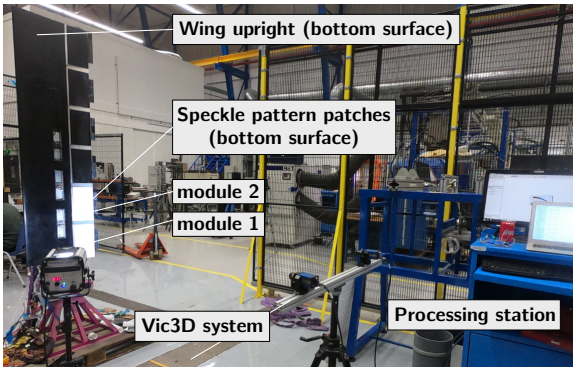


Figure II.1.16: Vic3D DIC measurements system.

II.1.4.3. Digital Image Correlation Setup

A Digital Image Correlation (DIC) static measurement was conducted on the top and bottom surfaces of the morphing modules to validate the design and assess the capability of the wing demonstrator to attain the static target morphing shapes,

Measurement Procedure

The DIC test setup is shown in Fig. II.1.16. The measurement system consisted of a Vic-3D stereo Q400 system equipped with a lens of 15 mm focal length [44]. The wing was placed upright, and the top and the bottom skin of two morphing modules (modules 1 and 2) were covered in a speckle pattern. The red patches in Fig. II.1.17 indicate the areas where the deformations were measured are indicated by the red patches in Fig. II.1.17. The trailing 130 mm portion and 190 mm portion of the wing's lower and upper surface were analysed. The measurement area spanned 200 mm on the top and 250 mm on the bottom surface in the spanwise direction, centred on the module.

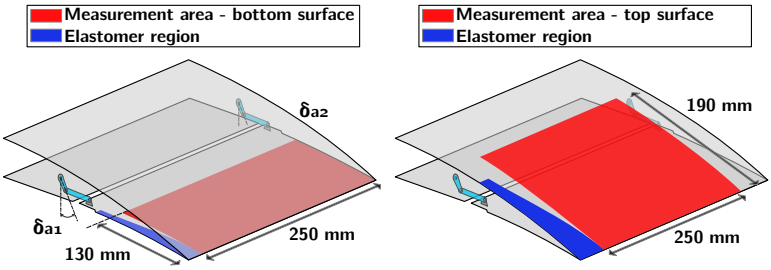


Figure II.1.17: Top and bottom DIC measurement area

A measurement test matrix was designed to reflect the morphing design limits of the airfoil (shown in Tab. II.1.3). Both modules' maximum allowable actuation limits were chosen between positive 25 and negative 25 degrees. The latter corresponded to the lower bound of approximately -4.5 mm chordwise travel (Bend Down) within the guided slot, as established by the FEM model. After evaluating the allowable stretch in the resulting

silicone skin for continuous operation, the upper bound (Bend Up) was also limited to 4.5 mm (25 degrees).

Table II.1.3: DIC measurement test matrix.

	Trial 1		Trial 2		
Condition	δ_{a1} [°]	δ_{a2} [°]	δ_{a1} [°]	δ_{a2} [°]	shape
S.Bend Down (-)	-25	-25	-25	-25	
S.Bend Up (+)	+25	+25	+25	+25	
A.Bend Down (+/-)	-25	-25	+25	+25	
A.Bend Up (+/-)	+25	+25	-25	-25	

The actuators were commanded in steps of 5 degrees according to the scheme presented in Tab. II.1.3. In this table, a single actuator's Bend Up and Bend Down deflection corresponds to +25 and -25 degree actuation inputs, respectively. In total, four cases were considered, (i) Bend Down, (ii) Bend Up, (iii) Differential Bend Down and (iv) Differential Bend Up. In the symmetric arrangement, both modules moved in the same direction (both up or down), and in the asymmetric case, the modules moved opposite to each other (one up and one down). The latter cases were designed to assess the impact of the elastomer skin on the morphed shape. Due to the differential actuation, stretching and shearing were expected to occur in the elastomer. The DIC static measurements were repeated twice for the symmetric case (trial 1 and trial 2) and compared with the prediction generated by the earlier numerical model.

The numerical model did not include the elastomer skin. Hence it is expected that the numerical results show the best possible morphing case, where the elastomeric skin does not affect the morphing displacement of the module. The measurements for the asymmetric case were conducted similarly in two trials and compared to the symmetric cases.

Calibration Procedure The DIC measurements were transformed into the same reference frame used in the numerical model to perform comparative analyses. This was achieved by orienting the trailing edge of the measured surfaces in line with the spanwise axis of the numerical model. A rotation about this axis was then performed to ensure the undeformed surfaces aligned with the undeformed analysis model. The DIC stereo camera setup was calibrated with a standard calibration target of a 30 mm circular grid pattern. The DIC calibration was verified on the top surface by comparing the trailing edge tip deflections measured with DIC against deflections measured with a Vernier height gauge.

Furthermore, a repeatability assessment was performed of the commanded morphed shape and the baseline shape at rest. The commanded actuator input configuration generated a repeatable morphed shape in both tests. Similarly, the baseline shape was assessed to ensure commanded shapes arrived at the same baseline in the unloaded case (at rest). This assessment was found to correspond well to the expected NACA 6510 airfoil.

II.1

II.1.5. Results and Discussion

The results and discussion are organised into three sections. First, the verification of the FSI tool and evaluation of non-linear structural effects are presented in Sec. II.1.5.1 and Sec. II.1.5.2. Then the outcome of the design optimisation is discussed in Sec. II.1.5.3. Finally, the validation of the morphing design is presented in Sec. II.1.5.4. In this last section, Digital Image Correlation (DIC) measurement was conducted on the top and bottom surfaces of the morphing wing to assess the capability of attaining static target shapes.

II.1.5.1. FSI Framework Verification

A key aspect in the FSI framework and, in particular, the system coupling between the FEM and aerodynamics was to determine: (i) the required resolution of both meshes to eliminate discretisation effects due to the selected mesh density and (ii) convergence criteria for the model to exit the FSI loop. Furthermore, an assessment was necessary of non-linear structural effects during large deformations. This was done by comparing linear and non-linear solutions in the FSI loop.

Model Convergence

The variation in deformations was analysed at each iteration, as was indicated previously in Fig. II.1.6, Sec II.1.3.1, to evaluate the convergence of the FSI loop. The change in the magnitude of deformation is calculated at each node, where the sum of these differences represents the total deformation variance of the system at each iteration.

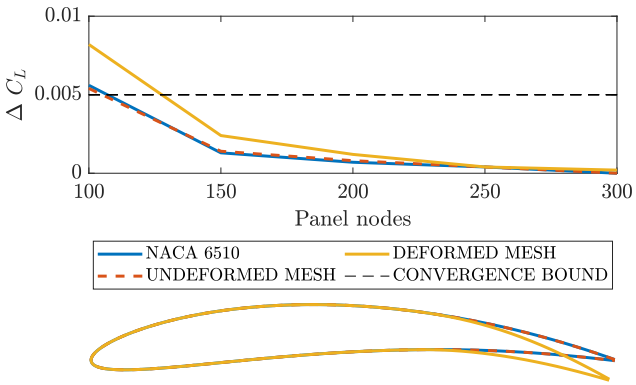


Figure II.1.18: Aerodynamic mesh convergence study of the lift coefficient [1].

Multiple analyses were run to a high iteration count with 2D XFOIL analysis to obtain a satisfactory convergence bound. The change in lift coefficient and actuator input loads was monitored between each iteration in addition to the deformation variance. It was found that when the total deformation variance of the system was below 0.01, the variation in the lift coefficient was less than 0.0001, representing 0.012% of the total C_L of baseline NACA6510. The actuation loads showed a variance of approximately 0.25% of peak load (approximately 0.15 N for all subcases)

With these negligible variations in lift and loads, the convergence performance was considered sufficient for the accuracy of this analysis, and the convergence bound was

set with a deformation variance of 10^-5. Setting this bound means that the simulation terminates once the deformation variance $\leq 10^-5$.

Mesh Convergence

A mesh convergence study was conducted on both meshes to determine a suitable resolution for the aerodynamic and structural mesh. The meshes were refined consecutively, starting from the aerodynamic mesh, followed by the structural mesh with the aerodynamic mesh fixed.

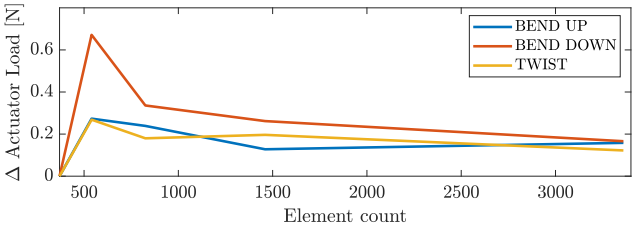


Figure II.1.19: Structural mesh convergence study [1].

For the aerodynamic mesh, three morphed states of the airfoil were analysed: (i) XFOIL's default NACA6510, (ii) undeformed structural mesh slice, and (iii) deformed structural mesh slice. Peak Bend Up (30 mm), Bend Down (20 mm), and Twist ($\pm$20 mm) were considered for the structural mesh study. The analysis was completed at zero angle of attack and a velocity of 30 m/s.

Results of the mesh resolution studies are presented in Fig. II.1.18 and Fig. II.1.19. For both meshes, convergence was evaluated by tracking the ΔC_L value. The convergence of the actuator input load was also assessed for the structural mesh.

As observed from Fig. II.1.18, the 2D aerodynamic mesh converged at 140 nodes in a chordwise direction. Therefore the default node XFOIL count of 160 nodes was maintained for future analyses. With this setting, the structural mesh converged beyond having 1500 elements. As the run time was sufficiently short, the mesh resolution was increased to apply ply dropping adequately, with more design freedom and accuracy. This was achieved by increasing the mesh seed size to 10 mm, corresponding to 3360 elements.

II.1.5.2. Evaluation of Non-Linear Solution

Table II.1.4: Comparison of linear and non-linear FEM model for given design cases [1].

Subcase	$C_{L_{\text{lin}}}$ [-]	$C_{L_{\text{non-lin}}}$ [-]	ΔC_L [%]	t_{lin} [s]	$\Delta t_{\text{non-lin}}$ [%]
Bend Up	-0.277	-0.296	-6.589	31.2	370
Bend Down	1.464	1.423	2.801	36.5	271
Twist	0.823	0.847	-2.916	42.7	165
Unmorphed	0.825	0.825	-	-	-

A further assessment was completed on the validity of a linear FEM model. Due to large deformations of the morphing surface, a comparison between linear and non-linear

solutions in the FSI loop is performed. In these analyses, the impact of the solver is studied for non-optimised skin. Peak Bend Up/Down and Twist cases were evaluated at zero angles of attack and wind speed of 30 m/s. Mesh densities were kept in accordance with the outcomes of the mesh convergence study.

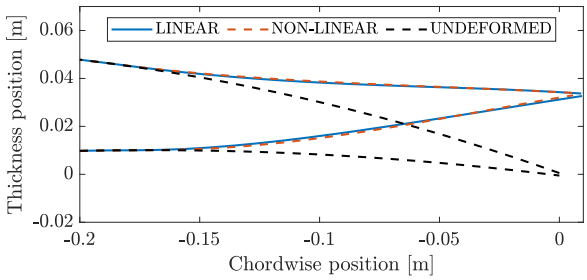


Figure II.1.20: Comparison of linear and non-linear deformation for the case Bend Up [1].

Comparison of lift coefficients is presented in Tab. II.1.4 [1]. Here in the last two columns, t_{lin} indicates the average execution times of linear solution in seconds, and $\Delta t_{\text{non-lin}}$, the average percentage increase of non-linear solution for each subcase. The benchmarking is performed over 10 FSI loops at 3 ° AoA and default meshing. A comparison of linear and non-linear deformations is presented in Fig. II.1.20, indicated by a blue and red-dashed line, respectively. The difference in deformation is minimal, as can be seen by overlapping blue and red-dashed lines; however, differences of 3-6 % are observed for the in C_L, with the highest difference in Bend Up case. Additional evaluation of actuator load for the linear and non-linear analysis revealed a comparable percentage difference, around three %.

It must be noted, however, that the execution times for non-linear cases presented in Tab. II.1.4, are significantly higher. The percentage increase in time versus linear is between 165-370%, which is undesirable for batch analysis. More importantly, the relatively small differences between linear and non-linear solutions will not significantly change the design regarding the required actuation load, as a margin is imposed for peak loads. This is discussed in the next section. Therefore, linear analysis is considered valid for design optimisation.

II.1.5.3. Design Outcome

The optimisation routine described in Sec. II.1.3.3 was performed for the subcases Bend Up, Bend Down, and Twist (Tab. II.1.2) to generate the final laminate design for the morphing concept. Furthermore, the FSI framework described in the design methodology in Sec. II.1.3 was used to finalise the actuator selection.

Laminate Design

During the optimisation process, a ply dropping sequence was determined. Fig. II.1.21 shows the undeformed morphing surface overlaid with the optimised thickness distribution. The colour map shows the number of plies required in each design region to build the morphing surface laminate.

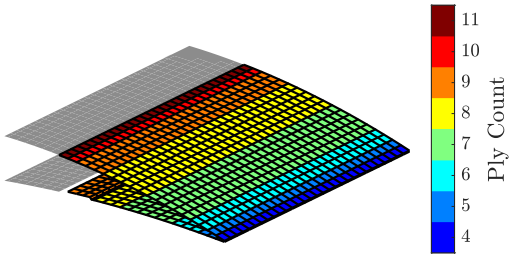


Figure II.1.21: Ply dropping sequence.

The deformed morphing surfaces for the three subcases are presented in Fig. II.1.22. They are superimposed on an undeformed surface to visualise the degree of deformations taking place. The colour map again represents the relative vertical deformations of the system.

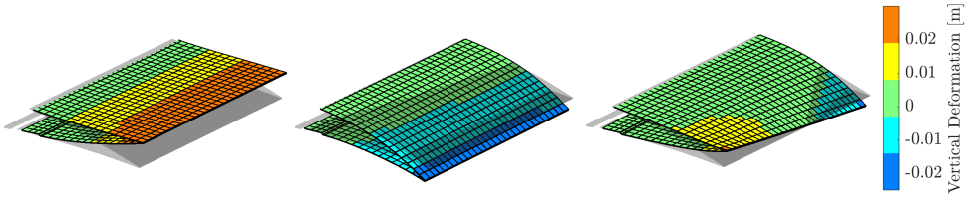


Figure II.1.22: Deformed morphing surfaces for subcases, Bend Up (left), Bend Down (middle) and Twist (right).

Actuator Loads

From the results of the optimisation routine, the peak actuation loads and the deflection range were determined. In Tab. II.1.5, the required input parameters for the three subcases are listed. Herein, the F_a and x_a indicate the actuation load and the horizontal deflection of actuator one, respectively.

Table II.1.5: Actuator peak forces for various subcases

Subcase	F_{a1} [N]	F_{a2} [N]	x_{a1} [mm]	x_{a2} [mm]	C_L [-]
Bend up	61	61	6.5	6.5	-1.7
Bend down	-60	-60	-4.5	-4.5	2.2
Twist	-50	23	4.5	-4.5	0.85

The actuation load ranges from -60 N to +61 N. The deflection range of the actuator ranges from -4.5 to +6.5 mm. The resulting lift coefficients range from -1.7 to +2.2. This information aided in the selection of a suitable actuator for the application.

The deflection range, x_a, of the free edge of the morphing surface is shown in Fig. II.1.23. With this information, the torque arm and rotation range, δ_{act}^{max}, of the servo could be set to calculate the torque requirements from the actuator. The actuator kinematic model was implemented according to the FBD illustrated in Fig. II.1.7.

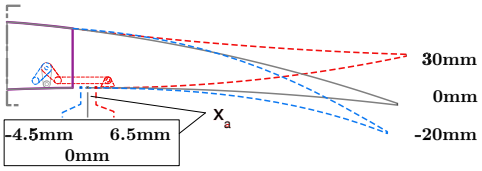


Figure II.1.23: Horizontal travel range.

The design aimed to minimise the torque arm and increase the available actuation force F_a. This also resulted in an increased required range of rotation to cover the entire horizontal travel range. The range of rotation was limited to 40° to ensure that the linkage forces remained in the linear range.

The relation of linkage force to δ_a^{max}, defined by the Eq. II.1.1, is presented in Fig. II.1.24 [1]. Herein, it can be observed that the ratio of the linkage force relative to the actuator force rapidly moves further away from the 1:1 ratio beyond 40° of rotation. A higher linkage to actuator force ratio is beneficial in actuation leverage. However, examining the FBD illustrated in Fig. II.1.7 it can be deduced that the larger ratio comes at the price of reduced linear travel for higher rotation angles; therefore, ±40° was maintained as the maximum rotation range. For this range of rotation, the torque arm required for 7 mm of horizontal travel was 10 mm.

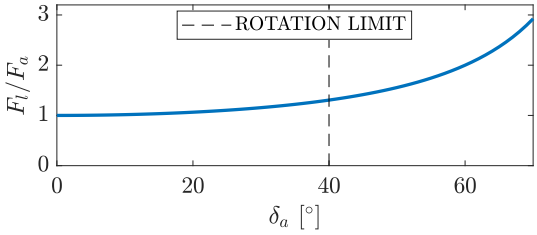
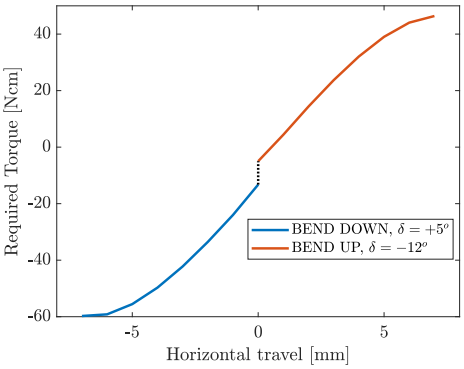


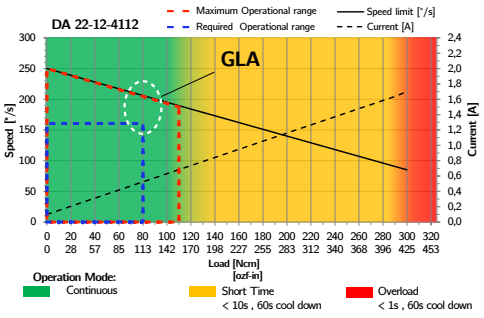
Figure II.1.24: Linkage force as a function of servo rotation [1].

The actuation loads were evaluated for the entire morphing range by incrementing horizontal travel at the actuation points, from -7 mm to 7 mm, in 1 mm increments to determine the actuator torque requirements. Only symmetric actuation was considered, as loads were most adverse in this scenario. The analysis was conducted at 30 m/s. To ensure most adverse aerodynamic loads were considered within the stall limits of the airfoil, the angle of attack was set to +5° for Bend Down cases and -12° for Bend Up cases. The data from these analyses are presented in Fig. II.1.25a. It can be seen that torque increases in a non-linear fashion, with peak torque requirements being nearly 45 Ncm for Bend Up and -60 Ncm for Bend Down. Note that, at zero travel, the lines do not coincide, as the analyses for the two cases were done at two different angles of attack.

For the peak loads presented in Tab. II.1.5, the peak torque required to actuate all morphing shapes at $V_\infty = 30$ m/s is ±60 Ncm. An operating margin of ≈ 35% of peak actuator torque was considered for continuous actuation at maximum free stream velocity and 1 Hz continuous sinusoidal actuation frequency at maximum rotation limits for the



(a) Torque versus horizontal travel, Bend Down and Bend Up considered [1].



(b) Volz da-22-12-4112 performance parameters in relation to control objectives [45].

Figure II.1.25: Actuator torque and performance parameters.

actuator selection. These requirements were selected to ensure the servo would not be used at 100% capacity during operation and would cope with the gust load alleviation task. These requirements resulted in ±80 Ncm torque and 160 °/sec continuous actuation at maximum load and maximum rotation limit (±40°).

With these two requirements, the selection was made between the available actuator models of the servo manufacturer Volz [45]. This servo supplier was chosen due to the real-time actuator load and position feedback needed for the controller design. Volz DA 22-12-4112 was selected as it met both the continuous torque and bandwidth requirements within the available range of actuators. Fig. II.1.25b shows the performance specification data of the actuator published by the manufacturer [45]. The green region indicates the continuously operational range of the servo. As seen, the required peak torque and the bandwidth indicated with a blue-dotted box fall within the servo's continuous operation range. At the peak torque of ±80 Ncm, the actuator is capable of delivering a continuous speed of 210°/sec, corresponding to tracking of 1.3 Hz sinusoidal signal at 40° peak rotations. This is indicated by the maximum operational limit, the red-dotted box. The effectiveness of the gust load alleviation can be the highest near the region indicated by the dotted white line (maximum load and speed).

II.1.5.4. Morphing Validation

A Digital Image Correlation (DIC) static measurement was conducted on the top and bottom surfaces of the morphing modules to validate the design of static target morphing shapes.

The DIC and the numerical model results were compared by comparing the TE tip deflection and the airfoil shape. First, the symmetric cases shall be discussed in Sec. II.1.5.4, where both modules moved in the same direction. Then, the mechanical losses in the morphing system are discussed in Sec. II.1.5.4, and reflections are made on the simulation model. Finally, in Sec. II.1.5.4 the asymmetric cases are discussed, where the modules moved opposite to each other and the impact of the elastomer is significant.

II.1

Morphing Assessment (Symmetric)

Figure II.1.26 shows tip deflection for the entire actuation range for two trials in both symmetric (red, blue), asymmetric cases (purple, yellow) and estimation of the numerical model (black dotted). Examining Fig. II.1.26 for the symmetric case, the immediate obser-

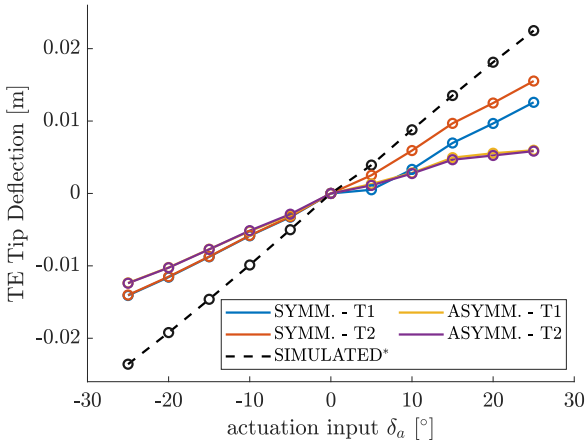


Figure II.1.26: Comparison of tip deflection between the numerical model and DIC measurement of symmetrical and asymmetrical cases $[25, -25]$ °.

vation yields that the linear curve predicted by the initial numerical model significantly overestimates the experimental tip deflection. This is observed more clearly in Fig. II.1.27 which shows a comparison of measured and simulated peak tip deflections (top row) and 3D shape (bottom row).

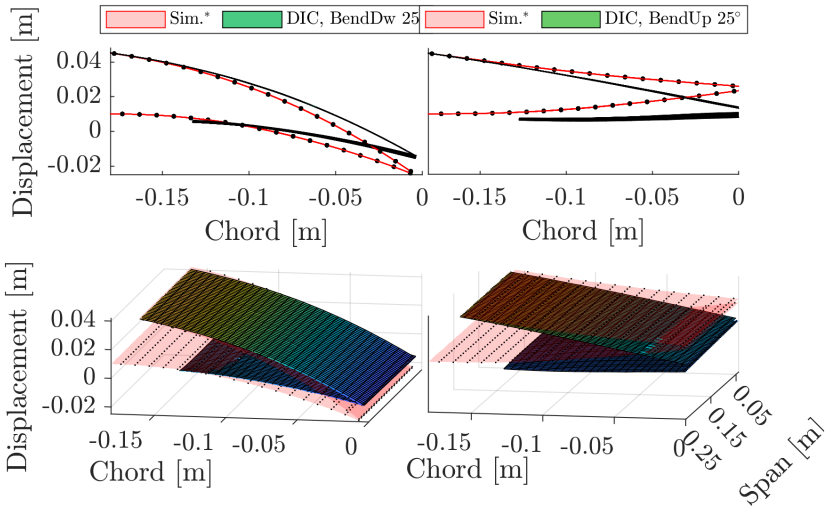


Figure II.1.27: Comparison DIC and initial simulation prediction peak Bend Up and down symmetrical cases.

The numerical model shows a linear slope between the actuator input range of ± 25 degrees. While this linear trend is upheld, in the 0 to -25 degree range, the experimental curves have a much shallower slope than the prediction. On the opposite interval, the linear trend exhibits a dead-band for the actuator displacement between 0 and 5 degrees in trials 1 and 2. The slope mismatch and the dead-band observations suggest that mechanical losses occur between the actuator input and the morphing shape, which the numerical model does not adequately capture. Consequently, the morphed trailing edge may fall short of delivering the maximum expected lift increment when subject to a free stream velocity. A more in-depth analysis was set up to investigate the mechanical losses in the system during actuation, explained in the following sections.

Mechanical Loss

After an initial evaluation of the numerical model, the mechanical system was observed more closely to investigate the potential source of mechanical losses during actuation. To this end, a visual inspection is performed during the Bend Up and Bend Down actuation strokes, and the image frames are processed with an image fusion algorithm shown in Fig. II.1.28.

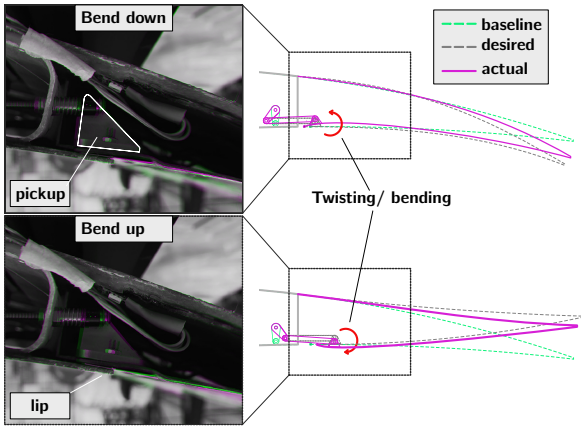


Figure II.1.28: Backlash effect observed in Bend Up and Bend Down.

In this figure, composite difference images are created of the image sequences at the maximum strokes of the actuator ($\pm 25^\circ$ actuator input), which correspond to Bend Up (bottom row) and Bend Down (top row) cases. The purple region indicates changes concerning the baseline position (zero $^\circ$ actuator input). The image frame captures the cross-section view of the mechanism, where the linkage arm is attached to the skin interface via a triangular pickup component. The purple areas clearly show the twisting of pickup and bulging in/out. The primary cause of this behaviour is attributed to the moment offset introduced as the servo arm pushes the vertex of the pickup and the relatively significant stiffness difference between the aluminium pickup body and the composite skin.

Observing the zoomed section of the bottom skin in Fig. II.1.29 measured with the DIC, the findings from the visual inspection are confirmed. Here, the out-of-plane deformation of the bottom section of the skin near the lip is shown in peak Bend Up and Bend Down

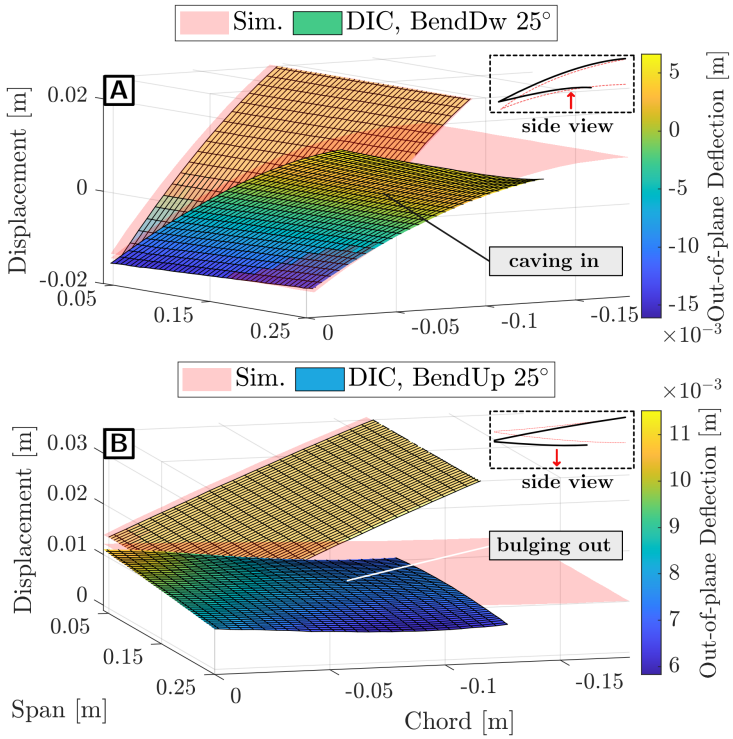


Figure II.1.29: Bottom skin behaviour during peak Bend Up and Down with DIC in 3D view.

cases, in Figs. II.1.29-A and II.1.29-B, respectively. Firstly, the lip bulges out during the bend upstroke and caves in during the Bend Down, reducing tip deflection. This behaviour is highlighted in the annotated cross-sectional view in Fig. II.1.29 for both cases. The bulging out effect is observed clearly in Fig. II.1.30, which shows the out-of-plane deflection in the 2D contour map for the Bend Up case in the top row of the figure. Secondly, from the 3D view of the Bend Up case, it is observed that the bottom skin deforms unevenly along the span. This is revealed in better detail in the contour map of the chordwise deflection shown in the bottom row of Fig. II.1.30. As seen, the isolines are skewed towards the one end of the module, and the magnitude of chordwise travel is nearly twice lower compared to the initial simulation.

Revised Model The undesired out-of-plane deformations observed in the measurements suggest that the actuator stroke is not fully converted to a chordwise translation of the skin; instead, a significant part of the stroke is lost in the twisting and bending of the bottom skin. From the perspective of the motion system, this behaviour can be characterised as backlash. This phenomenon is attributed to clearance inaccuracies and lost motion in mechanical systems, which can be partially compensated by increasing the actuator stroke. However, the numerical model cannot adequately capture this complex behaviour and predict the tip deflection due to inaccuracies.

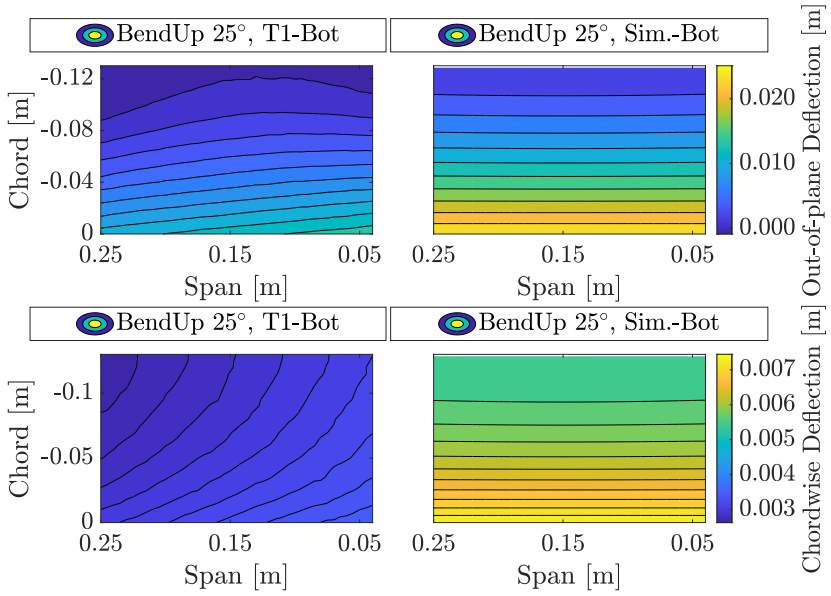


Figure II.1.30: Comparison of symmetric Bend Up case for the bottom surface.

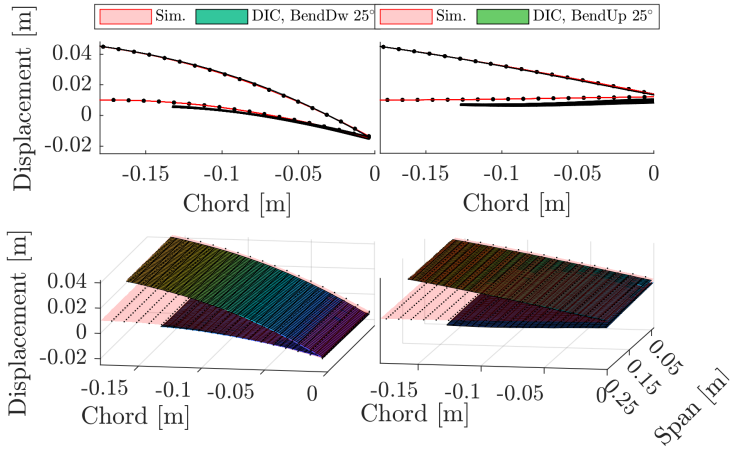


Figure II.1.31: Comparison DIC and simulation peak Bend Up/Down symmetrical cases with corrected mapping.

Adopting greater detail of actuation components and interfaces in the numerical model, e.g. accounting for guiding lip and the servo pickup, is a potential approach that can better estimate tip deflection. However, this will come at a high computational cost. Since the measured mapping follows a reasonably linear trend, shown in Fig. II.1.26, a cost-efficient way to improve the prediction at peak input can be done by adjusting the linear mapping between the actuator input and tip deflection. Figure. II.1.31 shows the comparison between the revised model and the DIC measurements in symmetric peak Bend Up and

Down. The revised model yields a good agreement in deflection. However, the bulging out effect cannot be captured, as shown in the airfoil shape in the top row. Consequently, this effect will be most apparent in the dead-band region, 0-5 degrees, where the linear trend does not hold, as shown in Fig. II.1.27.

It must be noted that the presence of aerodynamic load may further contribute to inaccuracies, which must be adequately accounted for during operation. Therefore, a potential solution is to include a compensator in the closed-loop control design, which can adjust the actuation input based on accurate instantaneous shape measurement. Robust, sensor-based control methods, such as the Incremental Nonlinear Dynamic Inversion (INDI), capable of mitigating model uncertainties through sensor feedback, are a suitable candidate [36]. In terms of sensors, embedded strain gauges, optical fibres, or camera tracking can be used [46, 47].

Morphing Assessment (Asymmetric)

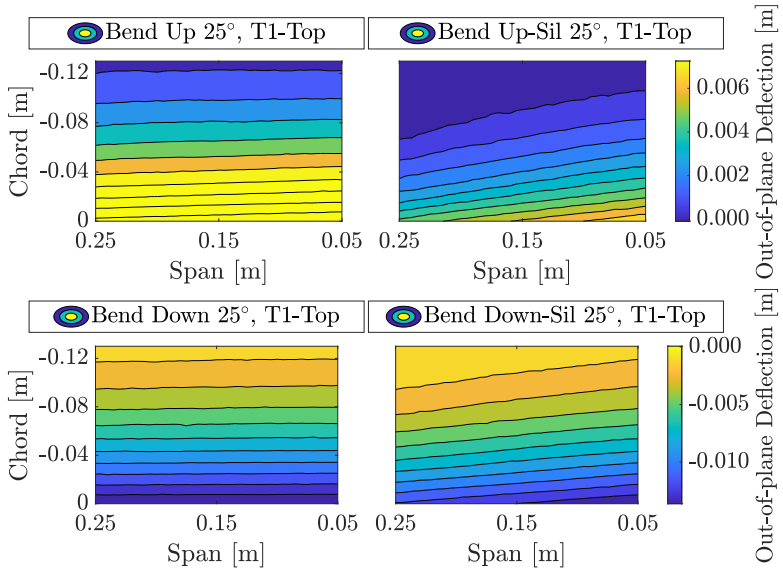


Figure II.1.32: Comparison asymmetric cases top surface.

The impact of the elastomer was assessed by evaluating the asymmetric cases measured by DIC, indicated by yellow and purple colours for trials 1 and 2, respectively, in Fig. II.1.26. The modules were actuated in the opposite direction, generating tension and shearing in the elastomer skin. No comparison is made with the numerical model, as the main subject of interest was to observe the impact of the silicone skin on the experimental model. In trials 1 and 2, backlash is observed, but the trials show good agreement. The main difference between the symmetric cases is lower tip deflection. In particular, the peak tip deflection is significantly lower (≈ 0.065 m versus ≈ 0.012 for trial 1) for the interval 0-25 degrees, corresponding to module 1 bending upwards and module 2 downwards.

Observing the out-of-plane deflection in Fig. II.1.32 for trial one on the top surface, Bend Down (bottom row) shows reasonable agreement; however, in the Bend Up case,

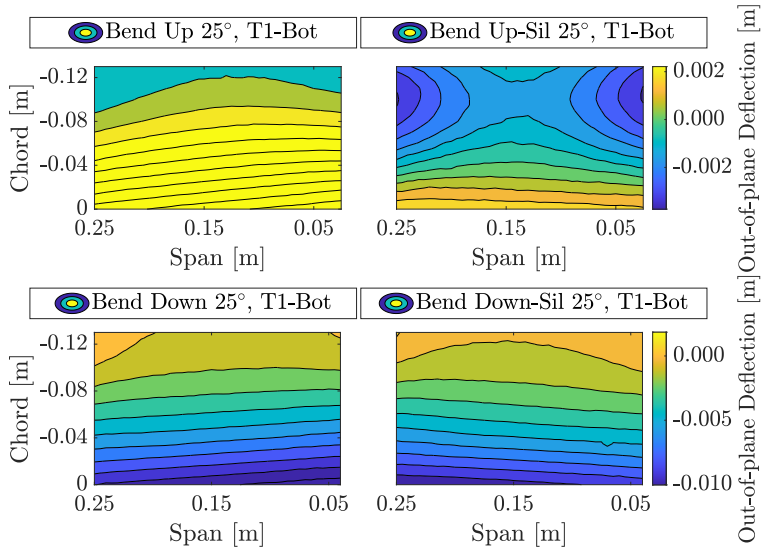


Figure II.1.33: Comparison of peak asymmetric cases bottom surface.

the deformations are significantly reduced. The same measurement on the bottom surface, shown in Fig. II.1.33, reveals a substantial bulging out effect; this is indicated with the highlighted annotation. This suggests that the actuation energy is consumed at peak deflection in bulging out of the skin rather than shearing the silicone skin. This also suggests that a better balance between the elastomer and wing skin stiffness is needed to allow more shearing in the critical connecting areas.

II.1.6. Conclusions and Outlook

A novel distributed morphing concept is presented, which addresses the drawbacks of the initial TRIC concept and allows variation of lift distribution locally along the span. The laminate design is optimised with an FSI optimisation framework considering the ply orientation, laminate thickness, laminate properties and actuation loads. Furthermore, a numerical and experimental study is performed to select a suitable elastomer design for inter-modular connection.

The desired morphing flexibility and manufacturability are met by a laminate design with fixed ± 45 degree ply orientation and gradual ply thickness dropping. The inter-modular connection is bridged with a wide silicon patch design, impacting the actuation loads and flexibility the least. The manufacturability of the concept is demonstrated by the realisation of the modular wing demonstrator. The finalised concept is validated for the commanded target shapes with DIC measurements and visual characterisation.

A linear trend is observed during symmetric Bend Up and Down actuation, agreeing with the simulation. However, the validation study of the concept revealed a significant sensitivity to the backlash in the sliding interface and added stiffness due to the addition of the elastomer. Consequently, the peak deflections are lower than anticipated from the

model. The measured input-output mappings are used to correct the model, improving peak deflection estimation. The model can be further improved with greater detail of the actuation components and interactions of the interfaces, such as the guiding lip and servo pickup.

However, a significantly more complex model is required to capture the combined effect of backlash accurately and added stiffness of the elastomer, particularly at the lower input range. Capturing and integrating these effects in the design process cost-efficiently presents a challenge worth investigating in further studies. Mitigating the backlash effect opens another opportunity for research that can be exploited with the current concept. A sensor-based compensator in the closed-loop controller, relying on accurate position measurements from the integrated sensors, is a possible approach. Finally, a further subject of ongoing research is to investigate the impact of a more flexible wing structure in torsion and spanwise bending.

REFERENCES

[1] T. Mkhoyan, N. R. Thakrar, R. De Breuker, and J. Sodja, "Design of a Smart Morphing Wing Using Integrated and Distributed Trailing Edge Camber Morphing," in *ASME 2020 Conference on Smart Materials, Adaptive Structures and Intelligent Systems; In Review at Smart Materials and Structures Journal*, ASME, sep 2020.

[2] D. Lentink, U. K. Müller, E. J. Stamhuis, R. De Kat, W. Van Gestel, L. L. Veldhuis, P. Henningsson, A. Hedenström, J. J. Videler, and J. L. Van Leeuwen, "How swifts control their glide performance with morphing wings," *Nature*, vol. 446, no. 7139, pp. 1082–1085, 2007.

[3] P. Henningsson, A. Hedenström, and R. J. Bomphrey, "Efficiency of lift production in flapping and gliding flight of swifts," *PLoS ONE*, vol. 9, no. 2, 2014.

[4] H. R. Jex and F. E. C. Culick, "Flight Control Dynamics of the 1903 Wright Flyer," in *12th Atmospheric Flight Mechanics Conference*, (Reston, Virigina), pp. 534–548, American Institute of Aeronautics and Astronautics, Aug. 1985.

[5] T. A. Weisshaar, "Morphing aircraft systems: Historical perspectives and future challenges," *Journal of Aircraft*, vol. 50, no. 2, pp. 337–353, 2013.

[6] S. Barbarino, O. Bilgen, R. M. Ajaj, M. I. Friswell, and D. J. Inman, "A Review of Morphing Aircraft," *Journal of Intelligent Material Systems and Structures*, vol. 22, no. 9, pp. 823–877, 2011.

[7] M. Kintscher, S. Geier, H. P. Monner, and M. Wiedemann, "Investigation of multimaterial laminates for smart droop nose devices," *29th Congress of the International Council of the Aeronautical Sciences, ICAS 2014*, pp. 1–11, 2014.

[8] J. Sodja, M. J. Martinez, J. C. Simpson, and R. De Breuker, "Experimental evaluation of a morphing leading edge concept," *Journal of Intelligent Material Systems and Structures*, vol. 30, no. 18-19, pp. 2953–2969, 2019.

[9] S. Vasista, A. De Gaspari, S. Ricci, J. Riemenschneider, H. P. Monner, and B. Van De Kamp, "Compliant structures-based wing and wingtip morphing devices," *Aircraft Engineering and Aerospace Technology*, vol. 88, pp. 311–330, mar 2016.

[10] F. Previtali, A. F. A. Arrieta, and P. Ermanni, "Performance of a Three-Dimensional Morphing Wing and Comparison with a Conventional Wing," *AIAA Journal*, vol. 52, pp. 2101–2113, Oct. 2014.

[11] G. Molinari, M. Quack, A. F. Arrieta, M. Morari, and P. Ermanni, "Design, realization and structural testing of a compliant adaptable wing," *Smart Materials and Structures*, vol. 24, no. 10, 2015.

[12] F. Previtali and P. Ermanni, "Performance of a non-tapered 3D morphing wing with integrated compliant ribs," *Smart Materials and Structures*, vol. 21, no. 5, 2012.

[13] G. Molinari, A. F. Arrieta, and P. Ermanni, "Aero-Structural Optimization of Three-Dimensional Adaptive Wings with Embedded Smart Actuators," *AIAA Journal*, vol. 52, pp. 1940–1951, Sept. 2014.

[14] B. K. S. Woods, I. Dayyani, and M. I. Friswell, "Fluid/Structure-Interaction Analysis of the Fish-Bone-Active-Camber Morphing Concept," *Journal of Aircraft*, vol. 52, pp. 307–319, Jan. 2015.

[15] B. K. Woods, O. Bilgen, and M. I. Friswell, "Wind tunnel testing of the fish bone active camber morphing concept," *Journal of Intelligent Material Systems and Structures*, vol. 25, pp. 772–785, May 2014.

[16] B. K. S. Woods, M. I. Friswell, and N. M. Wereley, "Advanced Kinematic Tailoring for Morphing Aircraft Actuation," *AIAA Journal*, vol. 52, pp. 788–798, Apr. 2014.

[17] S. Kota, J. A. Hetrick, R. Osborn, D. Paul, E. Pendleton, P. Flick, and C. Tilmann, "Design and application of compliant mechanisms for morphing aircraft structures," in *Smart Structures and Materials 2003: Industrial and Commercial Applications of Smart Structures Technologies* (E. H. Anderson, ed.), vol. 5054, p. 24, SPIE, aug 2003.

[18] S. Kota, P. Flick, and F. Collier, "Flight testing of the FlexFloil™ adaptive compliant trailing edge," *54th AIAA Aerospace Sciences Meeting*, 2016.

[19] B. Jenett, S. Calisch, D. Cellucci, N. Cramer, N. Gershenfeld, S. Swei, and K. C. Cheung, "Digital Morphing Wing: Active Wing Shaping Concept Using Composite Lattice-Based Cellular Structures," *Soft Robotics*, vol. 4, no. 1, pp. 33–48, 2017.

[20] N. B. Cramer, D. W. Cellucci, O. B. Formoso, C. E. Gregg, B. E. Jenett, J. H. Kim, M. Lendraitis, S. S. Swei, G. T. Trinh, K. V. Trinh, and K. C. Cheung, "Elastic shape morphing of ultralight structures by programmable assembly," *Smart Materials and Structures*, vol. 28, mar 2019.

[21] D. Keidel, U. Fasel, and P. Ermanni, "Concept Investigation of a Lightweight Composite Lattice Morphing Wing," *AIAA Journal*, vol. 59, pp. 2242–2250, jan 2021.

[22] O. Bilgen and M. I. Friswell, "Piezoceramic composite actuators for a solid-state variable-camber wing," *Journal of Intelligent Material Systems and Structures*, vol. 25, no. 7, pp. 806–817, 2014.

[23] O. Bilgen, L. M. Butt, S. R. Day, C. A. Sossi, J. P. Weaver, A. Wolek, W. H. Mason, and D. J. Inman, "A novel unmanned aircraft with solid-state control surfaces: Analysis and flight demonstration," *Journal of Intelligent Material Systems and Structures*, vol. 24, no. 2, pp. 147–167, 2013.

[24] A. M. Pankonien, C. T. Faria, and D. J. Inman, "Synergistic smart morphing aileron: Experimental quasi-static performance characterization," *Journal of Intelligent Material Systems and Structures*, vol. 26, no. 10, pp. 1179–1190, 2015.

[25] M. Mistry and F. Gandhi, "Design, fabrication, and benchtop testing of a helicopter rotor blade section with warp-induced spanwise camber variation," *Journal of Intelligent Material Systems and Structures*, vol. 26, no. 10, pp. 1272–1289, 2015.

[26] N. Nguyen, S. Lebofsky, E. Ting, U. Kaul, D. Chaparro, and J. Urnes, "Development of Variable Camber Continuous Trailing Edge Flap for Performance Adaptive Aeroelastic Wing," pp. 2015–01–2565, Sept. 2015.

[27] Y. Ferrier, N. T. Nguyen, E. Ting, D. Chaparro, X. Wang, C. C. de Visser, and Q. P. Chu, "Active Gust Load Alleviation of High-Aspect Ratio Flexible Wing Aircraft," in *2018 AIAA Guidance, Navigation, and Control Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, Jan. 2018.

[28] N. Nguyen, E. Ting, D. Chaparro, M. Drew, and S. Swei, "Multi-objective flight control for drag minimization and load alleviation of high-aspect ratio flexiblewing aircraft," in *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2017*, p. 1589, 2017.

[29] N. Werter, J. Sodja, G. Spirlet, and R. De Breuker, "Design and Experiments of a Warp Induced Camber and Twist Morphing Leading and Trailing Edge Device," Jan. 2016.

[30] X. Wang, T. Mkhoyan, and R. De Breuker, "Nonlinear Incremental Control for Flexible Aircraft Trajectory Tracking and Load Alleviation," *Journal of Guidance, Control, and Dynamics*, vol. 1, pp. 1–19, aug 2021.

[31] R. De Breuker, T. Mkhoyan, N. Nazeer, V. L. Stuber, R. M. Groves, S. van der Zwaag, and J. Sodja, "Overview of the SmartX wing technology integrator," *Submitted to AIAA Journal*, 2021.

[32] R. Vos, Z. Gürdal, and M. Abdalla, "Mechanism for Warp-Controlled Twist of a Morphing Wing," *Journal of Aircraft*, vol. 47, pp. 450–457, Mar. 2010.

[33] M. C. Drew, K. E. Hashemi, N. B. Cramer, and N. T. Nguyen, "Multi-objective Optimal Control of the 6-DoF Aeroservoelastic Common Research Model with Aspect Ratio 13.5 Wing," in *AIAA Scitech 2019 Forum*, (San Diego, California), American Institute of Aeronautics and Astronautics, Jan. 2019.

[34] M. d. F. V. Pereira, I. Kolmanovsky, C. E. S. Cesnik, and F. Vetrano, "Model Predictive Control Architectures for Maneuver Load Alleviation in Very Flexible Aircraft," in *AIAA Scitech 2019 Forum*, (San Diego, California), American Institute of Aeronautics and Astronautics, Jan. 2019.

[35] M. Dillsaver, C. E. S. Cesnik, and I. Kolmanovsky, "Trajectory Control of Very Flexible Aircraft with Gust Disturbance," in *AIAA Atmospheric Flight Mechanics (AFM) Conference*, (Boston, MA), American Institute of Aeronautics and Astronautics, Aug. 2013.

[36] X. Wang, E. Van Kampen, Q. P. Chu, and R. De Breuker, "Flexible Aircraft Gust Load Alleviation with Incremental Nonlinear Dynamic Inversion," *Journal of Guidance, Control, and Dynamics*, vol. 42, pp. 1519–1536, July 2019.

[37] W. P. Rodden and E. H. Johnson, *MSC/NASTRAN aeroelastic analysis: user's guide; Version 68.* MacNeal-Schwendler Corporation, 1994.

[38] M. Drela, "XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils," in *Low Reynolds Number Aerodynamics*, vol. 54, pp. 1–12, Berlin, Heidelberg: Springer Berlin Heidelberg, 1989. Series Title: Lecture Notes in Engineering.

[39] D. F. Watson, *Contouring: A Guide to the Analysis and Display of Spatial Data.* No. vol. 10 in Computer methods in the geosciences, Oxford ; New York: Pergamon Press, 1st ed ed., 1992.

[40] MSC Software Corporation, "MSC Nastran 2012 Design Sensitivity and Optimization User's Guide," tech. rep., MSC Nastran, 2011.

[41] Interglas, "Glass Filament Fabrics for Plastics Reinforcement - 92110 Product Specification," Tech. Rep. 0, Interglas, 2010.

[42] C. Thill, J. Etches, I. Bond, K. Potter, and P. Weaver, "Morphing skins," *Aeronautical Journal*, vol. 112, no. 1129, pp. 117–139, 2008.

[43] Wacker, "ELASTOSIL ®E41 TRANSPARENT," tech. rep., Wacker, 2020.

[44] Correlated Solutions, "VIC-3D 9 with iris System Specifications VIC-3D 9 System Specifications," tech. rep., Correlated Solutions, 2020.

[45] Volz, "Volz DA 22 Technical Specification," tech. rep., Volz Servos GmbH, 2013.

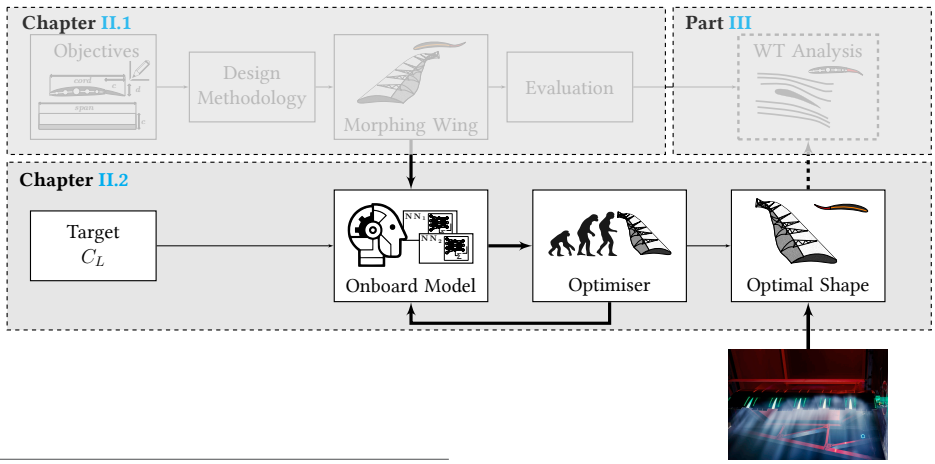
[46] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive Real-Time Clustering Method for Dynamic Visual Tracking of Very Flexible Wings," *Journal of Aerospace Information Systems*, vol. 18, pp. 58–79, jan 2021.

[47] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive State Estimation and Real-Time tracking of Aeroelastic Wings with Augmented Kalman filter and Kernelized Correlation Filter," in *AIAA Scitech 2021 Forum*, (Virtual Event), American Institute of Aeronautics and Astronautics, Jan. 2021.

II.2

Learning-Based Aerodynamic Performance Optimisation – Numerical

Building further upon the previous Chapter, this Chapter investigates how the aerodynamic performance can be optimised numerically, using the proposed seamless, distributed morphing wing design and artificial intelligence as a black-box controller. It aims to maximise the steady-state lift-to-drag ratio for a given target lift coefficient using lift and drag model predictions. The proposed method integrates an online-trained Artificial Neural Network (ANN) onboard model with an evolutionary optimisation algorithm [1].



This Chapter is based on the following peer-reviewed conference and journal paper:
O. Ruland, T. Mkhoyan, R. De Breuker, and X. Wang, "Black-box Online Aerodynamic Performance Optimization for a Seamless Wing with Distributed Morphing," *AIAA SciTech Forum 2022; In Review at Journal of Guidance, Control, and Dynamics*, p. 1840, 2022

Morphing is a promising bio-inspired technology with the potential to make aircraft more economical and sustainable by adapting the wing shape for the best efficiency in any flight condition. This Chapter proposes an online black-box performance optimisation strategy for a seamless wing with distributed morphing control. Pursuing global performance, the presented method integrates a global Radial Basis Function Neural Network (RBFNN) surrogate model with a derivative-free evolutionary optimisation algorithm. The effectiveness of the optimisation strategy was validated on a Vortex Lattice Method (VLM) aerodynamic model of an over-actuated morphing wing augmented by wind tunnel experiment data. Simulations show that the proposed method can control the morphing shape and angle of attack to achieve various target lift coefficients with better aerodynamic efficiency than the unmorphed wing shape. The global nature of the onboard model allows the presented method to find shape solutions for a wide range of target lift coefficients without the need for additional model excitation manoeuvres. Compared to the unmorphed shape, up to 14.6 % of lift-to-drag ratio increase is achieved.

Nomenclature

Symbols

A	=	amplitude
AR	=	aspect ratio
C_L	=	lift coefficient
$C_{\mathrm{L_t}}$	=	target lift coefficient
C_D	=	drag coefficient
$C_{\mathrm{D_0}}$	=	zero-lift-drag coefficient
C_J	=	cost penalty constant
c	=	chord
D	=	drag force
D_i	=	induced drag force
e	=	Oswald efficiency factor
f	=	frequency
J	=	cost
k_1	=	cost function singularity prevention quantity
k_2	=	cost function scaling factor
L	=	lift force
S	=	power spectral density
T_i	=	i^{th} Chebyshev polynomial
u_i	=	i^{th} virtual input
W	=	neural net weights
X	=	buffer model inputs
x_{hinge}	=	hinge location as a fraction of chord length
x_0	=	initial solution point
Y	=	buffer model outputs
y	=	spanwise coordinate
Z	=	frequency domain signal
z_{te}	=	trailing-edge displacement
α	=	angle of attack
Δ_f	=	change in frequency
δ_f	=	flap deflection
θ	=	actuator angle

σ_0	=	initial standard deviation
ϕ	=	phase

Subscripts

i	=	iteration
m	=	measured
s	=	saturated

II.2.1. Introduction

Over the past century, aircraft have become increasingly more efficient. During the 1960s, improvements in engine technology and wing design lead to significant improvements in aircraft fuel economy. In recent years, this trend of increasing efficiency has started to stagnate. More radical departures from conventional aircraft design are needed further to reduce the cost of flying and environmental pollution. One promising technology is active morphing, which enables shape transformation in-flight [2, 3]. The Wright Flyer, the first successful heavier-than-air powered aircraft, relied on twist morphing of its fabric-wrapped flexible wings to achieve roll control [4]. However, as aircraft flew at ever-increasing speeds, higher wing rigidity was required to sustain the loads, making morphing challenging with the available technology. In recent years, morphing has again been made possible by advanced developments in material science such as shape memory alloys, compliant mechanisms, and piezoelectrics [2, 5].

The ability to reshape the wing in flight introduces the problem of determining what that shape should be for a wide range of operational conditions. The current method for cruise drag minimisation is scheduling configuration settings through lookup tables as a function of gross weight, airspeed, and altitude. These lookup tables generally depend on analytical models validated with wind tunnel or test flight data. However, different operating conditions, aircraft production variances, and repairs can result in uncertainties in the table-lookup method.

Online optimisation can potentially tailor the wing shape to any specific flight condition for achieving the best aerodynamic performance based on in-flight measurements. Like birds, a smart morphing-wing aircraft could sense its environment and adapt its wings' shape to achieve the best performance in any condition, making it mission-adaptive. However, many challenges remain on the path toward operational smart morphing aircraft wings. To begin with, any online optimisation method relies on accurately evaluating the aircraft's performance using onboard sensors. Furthermore, only a very limited amount of search space exploration could realistically be afforded on a typical commercial flight. Ideally, a global optimum in the optimisation landscape should be found with limited and local explorations.

A real-time adaptive least-squares drag minimisation approach has been proposed for the variable camber continuous trailing edge flap (VCCTEF) described in [6, 7]. This strategy uses a recursive least squares algorithm to estimate the derivatives of the aerodynamic coefficients concerning the system inputs. The optimal wing shape and elevator deflection are then calculated from a constrained optimisation problem using the Newton-Raphson method. Improvements to the model excitation method, onboard model, and optimisation methods were demonstrated in wind tunnel experiments to achieve up to 9.4 % drag reduction on the common research model (CRM) with the VCCTEF at off-design conditions

at low subsonic speeds [8]. Simulations have also indicated that a 3.37 % drag reduction is achievable on the CRM with a distributed mini-plain flap system at Mach 0.85 [9].

Whilst the coefficients of the linear-in-the-parameters multivariate polynomial model adopted in [8, 9] can be estimated with relatively low computational cost, the model is only valid in the local region around the trim condition. This means that to perform real-time drag minimisation across the entire flight envelope, the model parameters must be re-identified at every operational point, or a global model must be constructed. Moreover, the required model excitation manoeuvres that comprise both angle of attack and flap deflection inputs would induce undesirable bumpiness, structural loads, and increased fuel consumption. Last but not least, using a local model together with a gradient-based optimisation method makes the solution prone to converge onto a local optimum. By contrast, while more difficult to identify online, a global onboard model could allow for continuous drag minimisation throughout the flight envelope. Additionally, when paired with a global optimisation method, global optima with even better performance could potentially be found.

The online performance optimisation strategy proposed in this Chapter integrates an online trained global artificial neural network (ANN) surrogate model [10], also referred to as the onboard model, with an evolutionary optimisation algorithm [11, 12]. The Covariance Matrix Adaptation – Evolutionary Strategy (CMA-ES) black-box optimisation method was adopted because of its robustness to noise, ability to optimise non-convex and multi-modal problems, and desirable global performance [13]. An onboard model is adapted online to reduce the time required for optimisation and effectively retain the knowledge gained from historical measurements. For the online identification of this onboard model, Radial Basis Function Neural Networks (RBFNNs) were employed because of their ability to absorb model updates locally, robustness to noise, and effectiveness on scattered data [14, 15]. Moreover, the black-box nature allows this type of network structure to be expanded easily to fit generic models or expand the domain of existing models. These characteristics make RBFNNs promising for in-flight aerodynamic model identification.

The integration of these methods allows for the optimisation of the morphing wing's shape based on scattered and noisy flight data in real-time. A high-level overview of the online optimisation framework is shown in Fig. II.2.1.

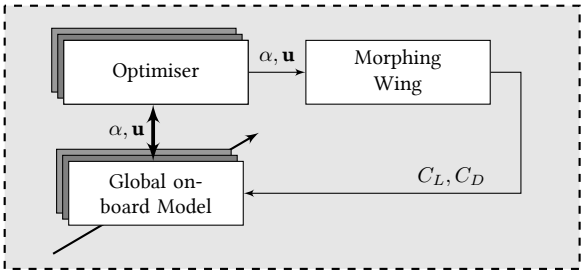
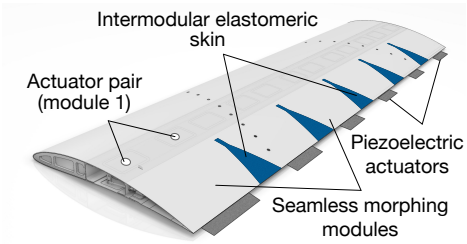


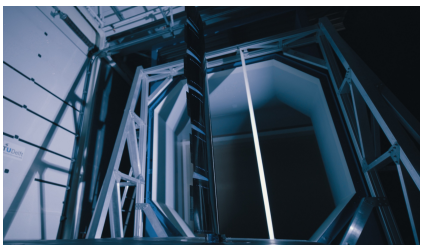
Figure II.2.1: High-level overview of the online shape optimisation framework.

During each iteration of the online optimisation method, the evolutionary optimiser uses the onboard model to evaluate the performance of populations of wing shape and

angle of attack combinations α, $\mathbf{u}$ to find the most promising input combinations. The most promising input combination is actuated and evaluated on the morphing wing system once per iteration. The resulting lift and drag coefficients C_L, C_D are then used to improve the onboard model for the next iteration. This online shape optimisation strategy is evaluated on a simulation model of an over-actuated seamless active distributed morphing wing named SmartX-Alpha [16]. An overview of SmartX-Alpha is shown in Fig. II.2.2a.



(a) Overview of wing components.



(b) Wind tunnel setup.

Figure II.2.2: Overview of the SmartX-Alpha wing demonstrator.

The 0.5 m $\times$ 1.8 m rectangular half wing is made up of six translation-induced camber (TRIC) morphing modules that comprise two actuators each and are seamlessly joined by a highly flexible elastomer skin. The TRIC morphing mechanism allows for both camber and twist morphing [17]. The ability to induce twisting of the trailing edge within each module by asymmetric actuator deflection allows for smoother morphing shapes than the distributed VCCTEFs. Control strategies for simultaneous gust and manoeuvre load alleviation have been demonstrated on SmartX-Alpha during wind tunnel experiments [18]. However, the in-flight drag minimisation using the distributed morphing of SmartX-Alpha remains an open challenge, which will be addressed in this Chapter.

The main contributions of this Chapter are the first presentation and demonstration of a novel adaptable in-flight black-box performance optimisation strategy for morphing wings. This data-driven approach is more adaptable and can potentially realise higher performance than conventional shape scheduling by lookup tables. The morphed wing shape could be tailored in flight to maximise the performance of the particular aircraft under consideration rather than the performance of a model built from previous test fight data on a similar aircraft. Moreover, compared to the state-of-the-art local grey-box methods, which require additional model excitation manoeuvres and re-identifications at each operational condition, the proposed approach retains the information learned in a global radial basis function neural network onboard model such that smooth and direct transitions to well-performing wing shapes can be achieved throughout the entire flight envelope. Furthermore, by integrating a derivative-free evolutionary optimisation strategy with a global onboard model, global optima can be found. The proposed method has been validated on a Vortex Lattice Method (VLM) model augmented by wind tunnel experiment data.

The structure of this Chapter is as follows. The morphing wing system is modelled in Sec. II.2.2. The optimisation architecture is proposed in Sec. II.2.3. The simulation results are presented and discussed in Sec. II.2.4. Finally, the main conclusions are drawn in Sec. II.2.5.

II.2.2. System Modelling

The following section discusses the system modelling approach utilised in this Chapter.

II.2.2.1. Virtual Inputs

The morphing wing system consists of 13 inputs: the deflections of the 12 actuators and the wing angle of attack. The optimisation algorithm is augmented with virtual shape functions to reduce the computational load of the optimisation problem while ensuring a smooth spanwise representation of the wing shape. This is achieved by using 5 virtual shape functions to describe the wing's shape instead of using the actuator angles directly as system inputs. These virtual inputs $u_1, ..., u_5$ scale the five basis shapes described by the first five Chebyshev polynomials of the first kind, re-scaled onto the $[0, 1.80]$ m domain, where 1.8 m is the half-wing span. Chebyshev polynomials were chosen as the parametric wing shape approximation function due to their nearly optimal property and orthogonality [19]. The spanwise distribution of the local actuator deflection is a linear combination of the virtual inputs and the Chebyshev polynomials $T_i(y)$ as stated in Eq. (II.2.1).

$$\theta(y) = \sum_{i=1}^{5} u_i T_i(y) \tag{II.2.1}$$

The order of the virtual shape function was chosen to be $q = 5$ because it resulted in the most significant reduction in computational loads without compromising the approximation power below acceptable levels for the expected wing shapes. Fig. II.2.3 demonstrates that the 5^{th} order Chebyshev polynomial has sufficient approximation for an elliptical distribution of the 1.8 m wing span.

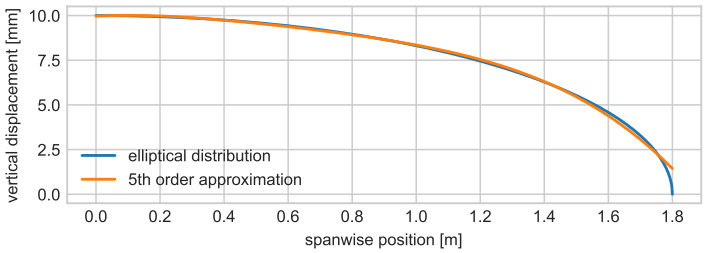


Figure II.2.3: Comparison of an elliptical distribution and 5^{th} order Chebyshev polynomial approximation.

Figure II.2.4 shows that most Root Mean Square Error (RMSE) reduction is achieved using the first five q polynomials, and the approximation RMSE is approximately 0.1 mm. In contrast, the reduction in RMSE for higher order ($q > 5$) stagnates, but the computational load increases significantly.

The resulting virtual inputs and their contributions to the actuator deflection at each location are shown in Fig. II.2.5, where the triangular markers indicate the actuator positions. The translation-induced camber morphing mechanisms are modelled as a series of twistable plain flaps, whose local deflections vary linearly between the actuators. The deflection of each actuator is, in turn, dictated by the virtual inputs.

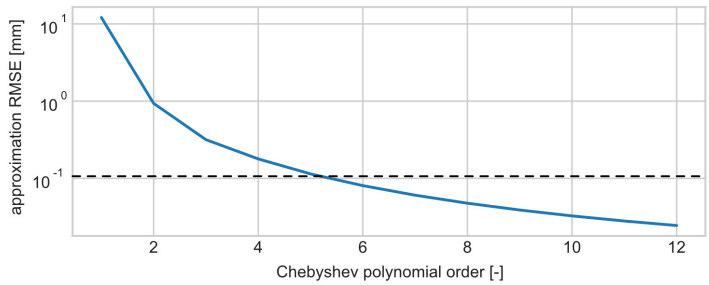


Figure II.2.4: Relation between the root mean square error of Chebyshev polynomials approximating an elliptical distribution and the model order.

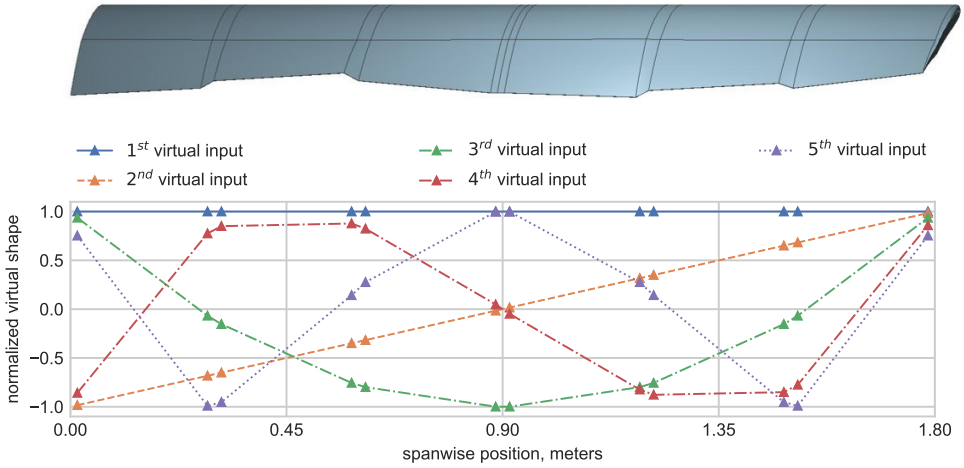


Figure II.2.5: Virtual shape functions that dictate the amount of camber morphing at each actuator location.

The virtual shapes reduce the 13-dimensional optimisation domain for the real system to a 6-dimensional one for the model. Concerning this cost of the optimisation framework, the overall benefit of using the virtual shape functions is threefold, namely, (i) a smaller, more efficient onboard RBFNN model can be used, which requires less time to train and update, (ii) the RBFNN model can be evaluated faster, and (iii) the computational cost to populate candidate solutions by CMA-ES is decreased. These aspects collectively reduce the overall optimisation cost of the framework. Moreover, the basis shape functions enforce a certain degree of smoothness in the final morphed wing shape. Their use generally leads to smoother shapes than those resulting from 12 independent actuator deflections as they avoid shapes with large and frequent jumps in spanwise camber. The choices for the optimiser and the onboard model are explained in Sections II.2.3.3 and II.2.3.2

II.2.2.2. Aerodynamic Model

The actuator deflections described by the virtual inputs are transformed into local flap deflections to produce the geometry to be evaluated by the aerodynamic model. First, the local vertical displacement of the trailing edge z_{te} is computed with Eq. (II.2.2), which was derived from digital image correlation measurements of symmetric morphing on SmartX-Alpha [16]:

$$z_{te} = \theta k_\theta \tag{II.2.2}$$

where $k_\theta = 5.6 \times 10^{-4}$.

The local plain flap deflection angle δ_f is then computed using Eq. (II.2.3), where x_{hinge} is the location of the flap hinge as a fraction of the chord length:

$$\delta_f = \sin^{-1}\left(\frac{z_{te}}{c \cdot (1 - x_{hinge})}\right) \tag{II.2.3}$$

Between the actuator locations, where the virtual inputs specify the local flap angle, the local flap angle varies linearly.

The aerodynamic performances of wing shape and angle of attack combinations are evaluated using a vortex lattice method (VLM) [20] model implemented in the Aerosandbox python package [21]. This method is used because of its high computational efficiency and scriptability. Since Aerosandbox is a relatively new open-source aerodynamic solver, and only one publication using this package exists in literature [22], its VLM implementation is verified against that of XFLR5 using the geometry of SmartX-Alpha. Figure II.2.6 shows results from the Aerosandbox and XFLR5 VLM solvers and wind tunnel measurements for a constant spanwise actuator angle of -22 degrees. It can be observed from Fig. II.2.6 that the outputs of Aerosandbox and XFLR5 VLM have a high consistency.

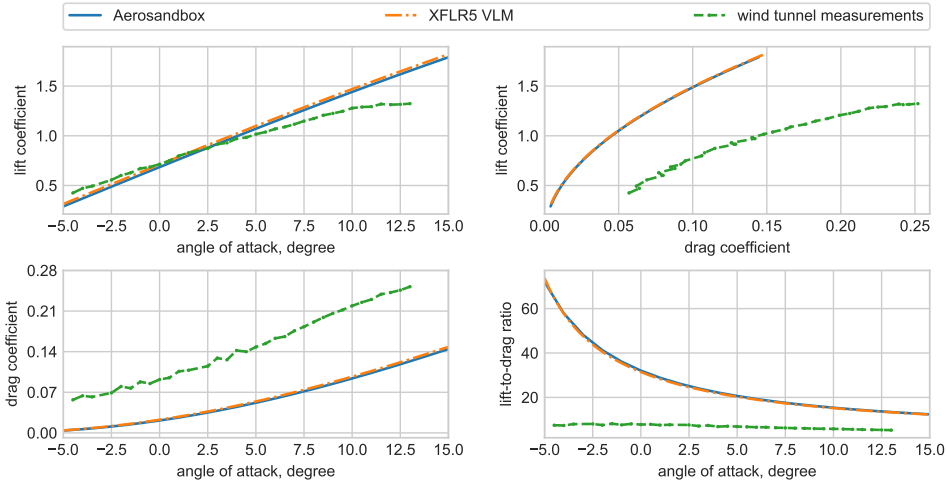


Figure II.2.6: Comparison of VLM solvers with wind tunnel measurements for a constant actuator angle of -22 degrees.

However, VLM neglects the effects of viscosity and thickness and can only be used to estimate lift and induced drag. As a result, the models slightly overestimate the lift slope, although their lift predictions remain close to the wind tunnel measurements for the linear part of the lift curve. On the other hand, drag is consistently underestimated due to the lack of viscous drag effects in the model. Furthermore, while asymmetric flap deflections affect the lift-to-induced-drag ratio $\frac{L}{D_i}$ through reshaping the spanwise lift distribution, constant flap deflections along the wingspan do not affect $\frac{L}{D_i}$ at all. However, to optimise the morphing wings' aerodynamic efficiency $\frac{L}{D}$, both the total drag and the effects of flap deflections on the lift-to-drag ratio should be modelled. Therefore, the model is augmented with an estimation of the zero-lift-drag coefficient C_{D_0} and a correction to the Oswald efficiency factor e based on data from a previous wind tunnel campaign with SmartX-Alpha. Furthermore, the use of the corrected model is restricted to the linear part of the lift curve, i.e., $-5.0 < \alpha < 10.0$ degrees. Wind tunnel measurements from seven angle of attack sweeps at different spanwise constant actuator angles were used to estimate C_{D_0} and e using the least-squares method and Eq. (II.2.4).

$$C_D = C_{D_0} + \frac{C_L^2}{\pi \mathit{AR} e} \tag{II.2.4}$$

The estimates for C_{D_0} and e were interpolated by 1st and 2nd order polynomials respectively, as shown in Fig. II.2.7. With these corrections and the induced drag from the Aerosandbox model, the total drag is estimated with Eq. (II.2.5), where $\bar{\delta}_f$ represents the mean flap angle. The efficiency factor of the constant deflection wing shape from Aerosandbox e_0 is estimated as 0.95.

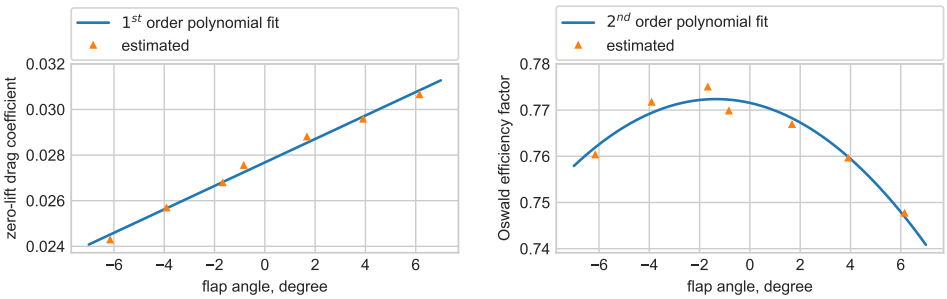


Figure II.2.7: Two correction functions estimated based on wind tunnel measurements.

$$C_D = C_{D_0}(\bar{\delta}_f) + C_{D_i} \cdot \frac{e_0}{e(\bar{\delta}_f)} \tag{II.2.5}$$

The effects of the corrections functions are shown in Fig. II.2.8 for the case of a constant -22 degree actuator angle. Compared with the uncorrected drag polar from Fig. II.2.6, the zero-lift-drag correction yields a much closer result to the wind tunnel measurements. However, the drag is still underestimated consistently. After correcting the drag predicted by Aerosandbox with the zero-lift-drag and the Oswald efficiency corrections, the resulting drag polar closely matches the wind tunnel measurements.

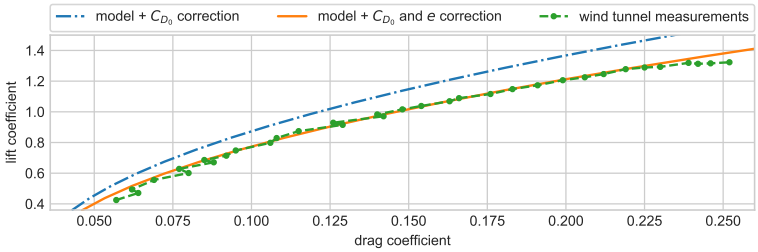


Figure II.2.8: Drag polar of the corrected aerodynamic model for a constant actuator angle of -22 degrees.

Since the corrections were estimated using wind tunnel data, their validity is limited to the wing geometry, and flow conditions to that these measurements correspond. In other words, the corrections in this section are only valid for the wing profile, the planform described above, and an airspeed of 15 m/s at sea level air density. Nevertheless, the presented correction method is widely applicable to other cases.

II.2.2.3. Secondary Model

For future real-world operations, white-box aerodynamic models such as the corrected model described above would be limited to training the onboard model beforehand. In this manner, a priori knowledge about the system is transferred to the onboard model through the network weights. Although these will be adjusted during the online learning process, fewer adjustments are required than would be in the case of learning from scratch. In later stages of the technology, the network weights would hold the knowledge from previous flights, which is superior in quality compared to any model-based predictions.

In order to demonstrate the ability of the online learning shape optimisation procedure to adapt to a change in the system to be optimised, a secondary aerodynamic model is used in the simulation of the online shape optimisation. The secondary model represents a comparable yet distinctly different morphing wing system.

The secondary model is comprised of the same wing planform as the nominal model but with a NACA4312 airfoil instead (the SmartX-Alpha airfoil is NACA6510). As the VLM solver does not model the effects of airfoil thickness, only the maximum camber and location of maximum camber are different between the nominal and secondary models. Because comparable wind tunnel data for this wing does not exist, the correction function estimation procedure cannot be repeated for the secondary wing model. Instead, the correction functions are altered directly. Therefore, the secondary model does not accurately model the aerodynamics of a known wing anymore. Instead, the second model represents the aerodynamics of an unknown wing, which are relatively close to those of the nominal model. The correction functions for both the nominal and secondary models are shown in Fig. II.2.9.

II.2.2.4. Noise Simulation and Filtering

Real-world measurements were simulated by adding noise to the aerodynamic model outputs. The noise realisations used were derived from noise measurements from a previous

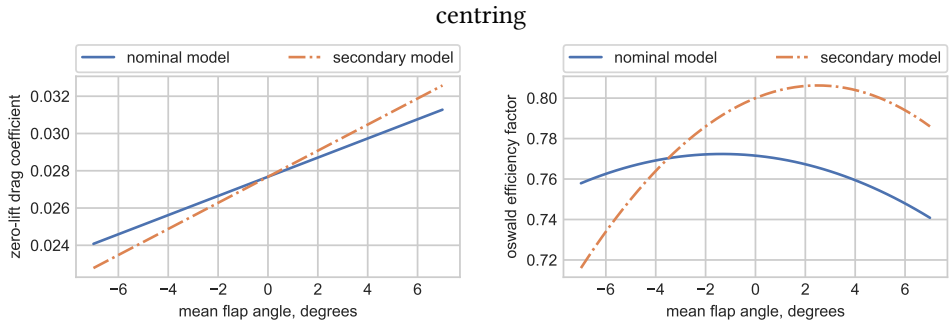


Figure II.2.9: Two correction functions for the nominal and secondary model.

wind tunnel experiment. The power spectral density (PSD) of the original noise signal, sampled at 1000 Hz, was approximated by its periodogram. The PSD $\mathbf{S}(\boldsymbol{f}_n)$ is sampled at n positive frequencies $\boldsymbol{f}_n = \begin{bmatrix}\Delta_f & 2\Delta_f & \dots & n\Delta_f\end{bmatrix}^T$.

First, these power spectral densities are converted to amplitudes using $\mathbf{A}(\boldsymbol{f}_n) = \sqrt{2\mathbf{S}(\boldsymbol{f}_n)}$, where $\mathbf{A}(\boldsymbol{f}_n)$ is the $n \times 1$ amplitude vector. Subsequently, the $n \times 1$ phase vector $\boldsymbol{\phi}(\boldsymbol{f}_n)$ is built by assigning each spectral component a random phase between 0 and 2π radians. Next, a frequency domain signal $\mathbf{Z}(\boldsymbol{f}_n)$ is constructed using Eq. (II.2.6).

$$\mathbf{Z}(\boldsymbol{f}_n) = \mathbf{A}(\boldsymbol{f}_n) \cdot e^{i\boldsymbol{\phi}(\boldsymbol{f}_n)} \tag{II.2.6}$$

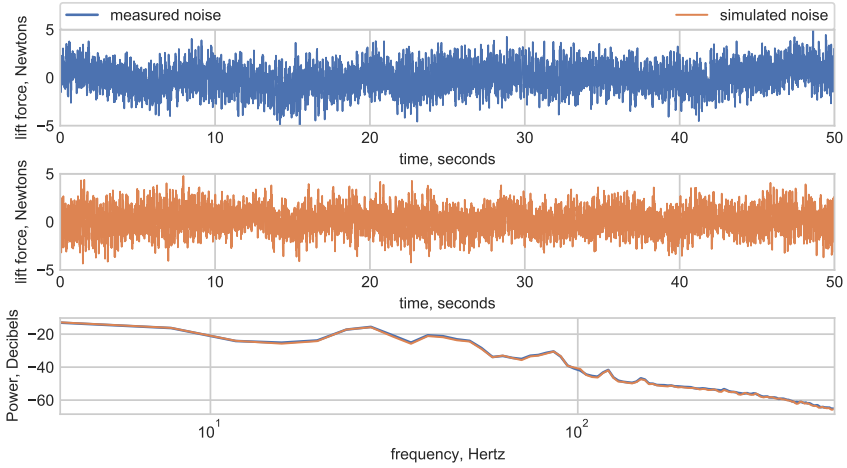


Figure II.2.10: Measured and simulated lift force noise signals.

Second, the frequency domain signal is transformed into a time-domain signal using the inverse fast Fourier transform. The resulting time-domain signal and the original noise measurement are shown in Fig. II.2.10. Although unique in the time domain, these noise

realisations are all made of the same spectral components. As such, the power spectral densities of both signals are nearly identical.

Finally, the system output measurements are simulated by averaging over the 50-second noise realisation for noise attenuation.

II.2.3. Optimisation Architecture

In this section, the online shape optimisation strategy and framework are proposed. First, an overview of the complete optimisation architecture is presented. Then each of the individual components is elaborated upon in the following subsections. The architecture of the proposed online shape optimisation framework is shown in Fig. II.2.11.

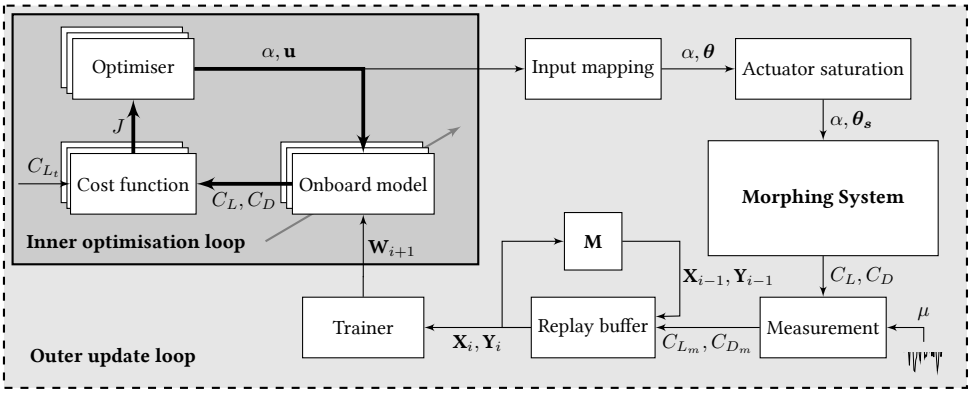


Figure II.2.11: Online shape optimisation architecture.

The optimisation procedure involves a fast and a slow loop. The optimiser, onboard model, and cost function work together in the last loop, marked by the shaded arrows. The optimiser evaluates angle of attack (α) and wing shape combinations ($\mathbf{u}$) on the onboard model with a high frequency. The resulting lift and drag coefficients from the onboard model are valued with a cost function (J), which is also based on the target lift coefficient (C_{L_t}). These cost values are, in turn, used by the optimiser to produce a more promising set of inputs for the next iteration of the optimisation loop. Once the optimiser has converged onto the most promising set of inputs, they are evaluated on the system itself in the outer loop.

The onboard model and optimiser describe the wing shape in 5 virtual inputs u_1, ..., u_5 for wing shape smoothness and computational load reduction. However, since the shape of the morphing wing is controlled by 12 actuators θ_1, ..., θ_{12}, the virtual input vector $\mathbf{u} \in \mathbb{R}^5$ needs to be mapped to the actuator input vector $\boldsymbol{\theta} \in \mathbb{R}^{12}$. Next, the actuator inputs are limited to their saturation limits of ± 25 degrees. Subsequently, the wing shape and angle of attack are actuated on the system. In this Chapter, the camber-morphing wing was simulated with an aerodynamic model of a wing with continuously distributed flaps. The resulting lift and drag coefficients are then contaminated with noise to simulate real-world measurements C_{L_m}, C_{D_m}. The inputs and outputs of the latest evaluation are added to the replay buffer, with a replacement strategy aimed at maintaining a global

coverage of the input domain in memory. The model inputs $\mathbf{X}_i$ and the model outputs $\mathbf{Y}_i$ in the buffer, making up the training set that is used to train the onboard model. The training of the artificial neural networks that make up the onboard model results in new network weights $\mathbf{W}_{i+1}$. From here on, a new optimisation cycle is initiated with an improved onboard model.

To evaluate the method's adaptability, weights from previous training on a different wing and no initial buffer data were used on the first iteration. In order to partly fill the empty buffer with data spread out over the input domain, the first 100 iterations were performed with quasi-random inputs instead of the optimiser-computed optima, known as the wandering phase.

As depicted in Fig. II.2.11, the optimiser does not work with the system directly but rather on the onboard surrogate model, which can be evaluated with much lower computational costs. The genetic optimisation algorithm queries the onboard model with a population of inputs to be evaluated. The quality of these inputs is then determined from the model's outputs using a cost function. The optimiser, in turn, uses this information to generate a new group of candidate solutions. This loop is continued until the optimiser converges, after which this most promising input can be tested on the actual system.

The objective of the optimiser is to find the set of inputs $\alpha, u_1, ..., u_5$ that maximises $\frac{C_L}{C_D}$ while meeting the target lift coefficient C_{L_t} without violating the angle of attack or actuator limits. The mathematical representation of this optimisation problem is:

$$\begin{aligned} \underset{\alpha,\mathbf{u}}{\textbf{MAXIMISE}} \quad & \frac{C_L(\alpha,\mathbf{u})}{C_D(\alpha,\mathbf{u})} \\ \textbf{SUBJECT TO} \quad & \alpha \in [\alpha_{\text{MIN}}, \alpha_{\text{MAX}}] \\ & \boldsymbol{\theta}_{\text{MIN}} < \boldsymbol{\theta}(\mathbf{u}) < \boldsymbol{\theta}_{\text{MAX}} \\ & C_L(\alpha,\mathbf{u}) = C_{L_t} \end{aligned} \tag{II.2.7}$$

This optimisation problem is nonlinear and non-convex because C_L and C_D are nonlinear and non-convex functions of α and $\mathbf{u}$.

II.2.3.1. Cost Function

As the optimiser queries the system with certain inputs, the corresponding outputs from the system need to be valued to, in turn, inform the optimiser how well the input is performed. The inputs cannot simply be scored on their associated drag, as this would tempt the optimiser into minimising the drag by minimising the lift produced. Instead, a promising angle of attack and wing shape combination should result in both a low drag coefficient and a lift coefficient close to the target lift coefficient. This is achieved with the cost function shown in Eq. (II.2.8).

$$J(C_L, C_D, C_{L_t}) = \underbrace{-\frac{C_L}{C_D}}_{\text{efficiency}} \cdot \underbrace{\frac{k_2}{k_1 + (C_L - C_{L_t})^2}}_{\text{deviation from lift target}} \tag{II.2.8}$$

The cost of any set of system outputs is dependent on the lift and drag coefficients and the target lift coefficient. The cost varies linearly with the aerodynamic efficiency $\frac{C_L}{C_D}$ and is inverse-quadratically related to the difference between the target and actual lift

coefficients. A small quantity $k_1 = 1 \times 10^{-4}$ is added to prevent singularities for small error values. The parameter $k_2 = 2 \times 10^{-5}$ is used to scale the output to $[-1, 0]$. Two and three dimensional plots of the cost function for $C_{L_t} = 0.50$ are shown in Fig. II.2.12. Note that the cost increases rapidly for any deviation from the target lift coefficient, while steps in the drag-coefficient axis generally result in smaller cost variations. In other words, a solution that provides low drag at a wrong lift coefficient is valued similarly to a solution associated with a higher drag at the right lift coefficient.

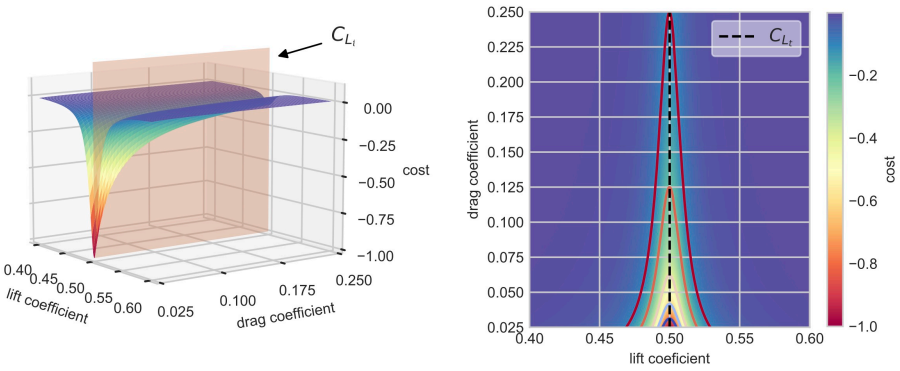


Figure II.2.12: Isometric (left) and top-down (right) view of the cost function for $\mathbf{C_{L_T} = 0.50}$.

Additionally, the angle of attack and actuator constraints are also handled by the cost function. If a set of inputs violates any of these constraints, its cost becomes, as shown by Eq. (II.2.9).

$$J = (\alpha_i - \alpha^\star)^2 + C_J, \quad J = (\theta_i - \theta^\star)^2 + C_J \tag{II.2.9}$$

In the case that the angle of attack of a set of inputs to be evaluated is outside the bounds $[-2.5, 10.0]$ degrees, the associated cost will be the square of the difference between the angle of attack α and the middle of the domain $\alpha^\star = 3.75$ degrees plus a large constant C_J. The valid range of θ is $[-25, 25]$ degrees so $\theta^\star = 0$ and Eq. (II.2.9) reduces to $J = (\theta_i)^2 + C_J$. This cost penalty constant is set to $C_J = 10$ to ensure that the cost will always be higher than that of an input set that is not in violation of these constraints. The square term serves to provide a gradient towards the middle of the parameter domain.

It is important to note that although the cost function is formulated as a non-linear convex cost function, the relationship of the predicted lift and drag coefficients is non-convex and complex. The latter parameters depend on the angle of attack, α, and the virtual shape inputs, $\mathbf{u}$. Therefore, although the relationship between C_L, C_L and J is convex, the relationship between α, u and C_L, C_L is non-convex. This necessitates using non-linear, non-convex optimisers capable of interacting with global models.

II.2.3.2. Optimiser

The optimiser aims to find inputs to the onboard model that minimise the cost of the model outputs as determined by the cost function. This optimisation is performed with

the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) algorithm [13]. CMA-ES is an evolutionary strategy for black-box optimisation of non-linear, non-convex, and continuous problems. It can handle multi-modality and discontinuities in the function to be optimised and has desirable global performance. CMA-ES has been used for offline aero-structural optimisation of a 3D morphing wing model in [23, 24].

In the framework proposed in this Chapter, CMA-ES operates by iteratively generating populations of inputs that are subsequently evaluated on the onboard model. Based on the returned costs of these candidate solutions, the mean and covariance matrix of the next generation's population are adapted. This process is repeated until the variation of the cost function converges to a threshold, selected as 1×10^{-6}. In the online shape optimisation procedure, a population size of 150 was used, referred to as parameter λ. The middle of the input domain was used as the initial solution point $\mathbf{x}_0$. Furthermore, the initial standard deviation $\boldsymbol{\sigma}_0$ and the scaling of the input variables were selected such that $\mathbf{x}_0 \pm 2\boldsymbol{\sigma}_0$ spanned the width of the inputs domains.

The total required number of function evaluations depends on the population size and also varies naturally due to the stochastic nature of the evolutionary strategy. Optimisation with higher population sizes generally requires fewer optimiser iterations and more system evaluations per iteration. With a population size of 150, on average, 180 optimiser iterations were needed, with a total number of system evaluations of 27,000.

The optimisation problem is required to optimise the angle of attack and virtual inputs for a given target lift coefficient while maximising the steady-state lift-to-drag ratio. Therefore, a limited but considerable time window could be reserved to find the solution iteratively. The selection of the population size was made in the effort to utilise the given time window for calculation fully. With the selected system and population size, convergence was achieved on average within 10.7s and 7.5s on Intel Core i7-4510U Central Processing Unit (CPU), 8.00 GB Random-access Memory (RAM) and Intel Xeon W-2223 CPU 3.60 GHz, 16.00 GB RAM, respectively. The robustness of the convergence of the optimiser with chosen population size was assessed by evaluating the offline trained onboard model for 100 runs for the target lift coefficient of $C_{L_t} = 0.50$. Figure. II.2.13 shows the variation of the six parameters, the angle of attack, and five virtual inputs. As shown in the figure, the variance of the five virtual inputs describing the optimised trailing edge displacements of the morphing wing is contained within 0.05 mm. Furthermore, the variation in the angle of attack is below 0.004 degrees. Collectively, these variations are considered sufficient for the given optimisation problem.

II.2.3.3. Onboard Model

The onboard model consists of two RBFNNs that model the mapping of the system inputs α, u_1, ..., u_5 to the lift and drag coefficients. The C_L and C_D networks consist of a single hidden layer with 500 and 940 centers, respectively. More approximation power is needed for the C_D network than for the C_L network because of the higher degree of nonlinearity of the drag relation compared to the lift relation. The choice for this RBFNN structure is made due to its ability to absorb model updates locally [14], and since it allows expanding the input space of the model with minimal changes to the remainder of the framework.

The training of the neural networks is done with mini-batch online training, with a batch size of 32. During training, the network weights are updated using the Adagrad

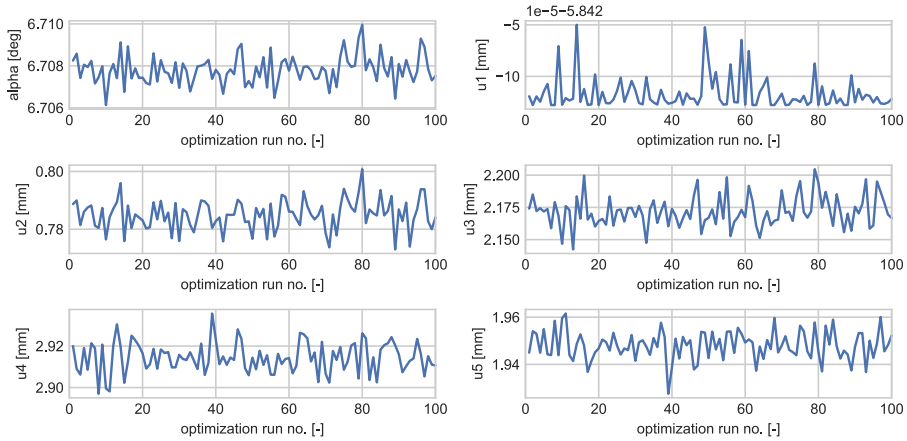


Figure II.2.13: Convergence results for 100 optimisation runs of the optimiser for the angle of attack and virtual inputs.

algorithm proposed by Duchi et al. [25], with an initial learning rate of 0.01 and a mean squared error loss function.

The neural network models are not initialised with random weights but rather with stored weights from a previous training session. In future applications, such a previous training session would be the online training performed during the most recent flight. For the simulations in this study, the starting weights for the online shape optimisation will be weights from offline training on the nominal aerodynamic model. It is noteworthy that the simulated online optimisation operates with the secondary model in the loop. Therefore, the initial weights serve only as a starting point and do not yet constitute a model representative of the system to be optimised.

For the initial offline training of the onboard model, a data set consisting of 261,360 wing shape and angle of attack combinations and their resulting lift and drag coefficients on the nominal model was used, with 10% of the data being reserved for validation. Both neural nets were trained from scratch for 2,000 epochs, which equated to roughly 23 hours of training time on a laptop (Intel® Core™ i7-4510U CPU, 8.00 GB RAM). Figure II.2.14 shows the corresponding training and validation losses, converted to Normalised Root Mean Square Errors (NRMSEs) for ease of comparison.

Even with the higher approximation power of the C_D network, the NRMSE of the C_L network is lower because of the lower degree of nonlinearity in the lift relation. Both networks' loss curves still show a decreasing trend towards the end of the training session. The training cut-off at 2,000 epochs is a trade-off between computational cost and starting point quality. The increased computational costs of further training yield an increasingly diminished return in accuracy, and the networks are only to serve as a starting point for the onboard model.

The low computational cost is the main benefit of using the onboard model instead of direct system evaluations. The CMA-ES optimiser typically requires thousands of function evaluations to converge on an optimum. In the neural network models, hundreds of input

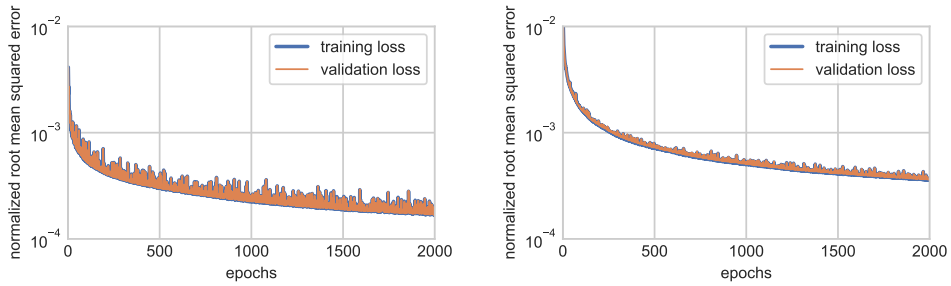


Figure II.2.14: Training and validation losses for the lift (left) and drag coefficient networks (right) in offline training.

combinations can be evaluated in less than one second, whereas in the aerodynamic model, each evaluation takes 1.5 seconds on average. In other words, the indirect optimisation using the onboard model is approximately 2500 times faster than the direct optimisation on the aerodynamic model.

On a real-world aircraft, considerably more time would be required because of transients and noise filtering, making direct optimisation unfeasible. Direct optimisation using the nominal aerodynamic model and indirect optimisation using the offline-trained onboard model were performed for several target lift coefficients. In order to make the computational time of the direct optimisation more feasible, a population size of 9 was used for both. The resulting optimal shapes as computed by the CMA-ES optimiser are shown in Fig. II.2.15.

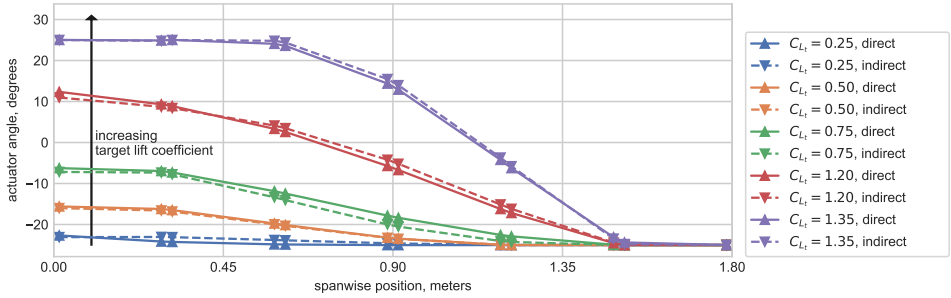


Figure II.2.15: Optimal wing shapes computed directly and indirectly on the system, for various target lift coefficients.

The optimal shapes computed by indirect optimisation are very close to those computed using the system directly. On average, the direct optimisation took 44.7 minutes per target lift coefficient, whereas the average computational time of the indirect optimisation was only 3.9 seconds (about 688 times faster).

II.2.3.4. Replay Buffer

During the online mini-batch training, the onboard model is trained on a training data set in memory in the replay buffer. This buffer consists of a history of evaluated inputs and their corresponding lift and drag measurements. Since the onboard model is adjusted to adapt to this data, the contents of the buffer are of critical importance. If the training data set lacks data points in a domain region, then the neural nets will unlearn the previously learned points in this region. This phenomenon, known as catastrophic forgetting, was first described in [26]. Therefore, a simple first-in-first-out training set buffer will not be sufficient to learn and retain a globally accurate onboard model.

Instead, the replacement of old data points when the buffer is full is based on the nearest neighbour search on all points in the buffer inspired by the coverage maximisation strategy described in [27]. The data point with the lowest mean euclidean distance to its 10 closest neighbours is replaced with the latest available data point. This replacement strategy aims to maximise the coverage domain of the training set by replacing the data points in regions of high data density and holding onto samples in data scare regions of the domain.

The maximum buffer size was not reached during the relatively short simulations presented in this study. In the future, the adaptation speed could be further improved by prioritising newer data points over older ones in the more densely populated domain areas.

II.2.3.5. Summary and Remarks

It is important to note that although the cost function is formulated as a non-linear convex cost function (see Fig, II.2.12 for $C_{L_t} = 0.50$), the relationship of the predicted lift and drag coefficients is non-convex, and depend on the angle of attack, α, and the virtual shape inputs, $\mathbf{u}$. For this reason, although the relationship between C_L, C_D and J is convex, the relationship between α, u and C_L, C_D is non-convex. This necessitates the use of non-linear, non-convex optimisers capable of optimising global models. Furthermore, given that the aerodynamic model on which the onboard model is trained does not hold explicit formulation of the candidate shape and angle of attack solutions, the candidate solutions are proposed to be evaluated iteratively in the inner optimisation loop. This brings the importance of the second part, namely the need for a low-cost data-driven onboard model which can be updated globally in an outer loop using the measurements from the actual shapes evaluated online. For this purpose, the low-cost RBFNN is proposed with local sensitivity to measurement updates. Finally, the global black-box nature and local sensitivity of the RBFNN are proposed as suitable candidates for implementing the online shape optimisation strategy in a real-life experiment using actual measurements. The implementation of this is discussed in Chapter III.3.

In summary, the optimisation architecture shown in Fig. II.2.11 two essential components, namely, the fast inner optimisation loop and the slower outer loop, which are responsible for updating the global RBFNN model based on new measurements of the lift and drag coefficient obtained from the optimised shapes produced by the aerodynamic wing model. The fast inner loop efficiently evaluates the candidate solutions of virtual shapes and angles of attack for a given target lift coefficient. Given the current optimisation structure with the fast inner optimisation loop and the low, the CMA-ES lends itself

suitable for implementing a parallel query approach. In this approach, the computational time can be further reduced by implementing a query of multiple candidate solutions on the low-cost RBFNN model. The actual approach using the experimental wing model is discussed in Chapter III.3.

II.2.4. Results and Discussion

In this section, the results from two simulation experiments are presented. The online optimisation algorithm was run for 15 iterations during the first simulation with a fixed target lift coefficient of 0.75. During the second simulation, 275 iterations of online shape optimisation were simulated with a target lift coefficient varying between 0.25 and 1.25. The aerodynamic efficiency of the resulting wing shapes is compared to that of the wing jig shape. The wing jig shape is defined as the shape of the wing at rest, with all morphing actuators set to zero deflection. The wing jig shape does not have any pre-twist.

II.2.4.1. Single Target Lift Coefficient

The online shape optimisation framework was run for 115 iterations, of which the first 100 were performed in wandering mode and the rest in optimisation mode. The inputs that were evaluated on the system are shown in Fig. II.2.16, where the optimisation phase is marked with a red background.

As expected, the angle of attack and the virtual inputs vary within their bounds with no recognisable pattern during the wandering phase. The cost associated with these pseudo-random inputs is generally high, except at iteration 26, where the resulting lift coefficient was relatively close to the target lift coefficient by coincidence. Shortly after the algorithm enters the optimisation phase at iteration 101, the inputs plateau. At iteration 102, a shape is tried, resulting in a higher cost than the shape from the previous iteration. Subsequently, the inputs move away from this location, and the associated cost decreases and converges.

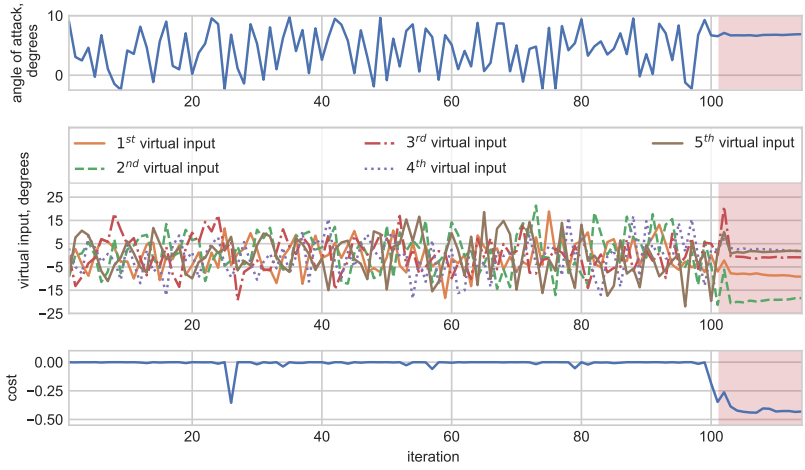


Figure II.2.16: Input history for wandering and optimisation (red background) with $C_{L_t} = 0.75$.

More insight into the inner mechanisms of the optimisation algorithm is provided by the optimal inputs as calculated by the optimiser, shown in Fig. II.2.17. The optimal angle of attack and optimal virtual inputs remain unchanged for the first 32 iterations of the wandering phase. During this period, measurements are collected, and the training buffer is partially filled. Training of the onboard model is only started after the size of the training set exceeds the batch size used for training.

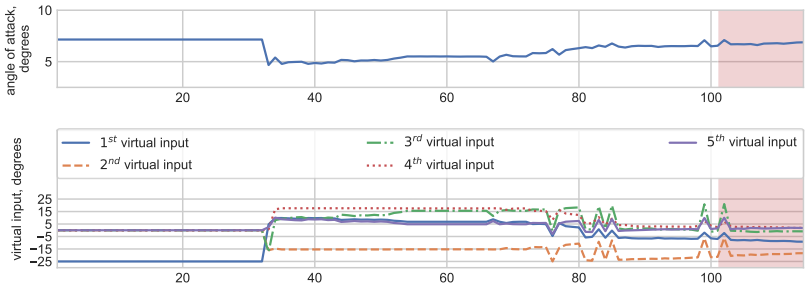


Figure II.2.17: Optimal inputs as calculated by the optimiser for a target lift coefficient of 0.75.

At iteration 32, the online training is started, and the algorithm's estimation of the optimal input changes with a sudden jump for the first time as the global minimum of the onboard model has shifted. Subsequently, the estimation of the optimal inputs changes repeatedly as the onboard model keeps training on increasing data points and starts to represent the system more accurately. The fact that the optimal inputs change slowly during the optimisation phase, where estimated optimal inputs are evaluated on the system, indicates that the onboard model has captured the trends in the exploratory data quite well during the wandering phase.

Two spikes in estimated optimal input can be observed at iterations 99 and 102. These two points correspond to an input that seemed promising based on the onboard model at the end of the wandering phase. However, the system yielded a lower performance than expected once tested. After evaluating the system, this input combination does not show up in the optimal inputs in later iterations.

The wing shapes evaluated on the system during both phases are shown in Fig. II.2.18. The pseudo-random shapes, shown in blue, span the entire actuator domain. The optimal wing shape, shown in orange, starts with only minor changes in camber near the wing root, compared to the wing's jig shape. Towards the tip of the wing, its camber is decreased until the actuators in the tip module hit their maximum negative deflection angles of -25 degrees. This morphing shape brings the spanwise lift distribution of this zero-twist rectangular planform wing closer to the theoretically ideal elliptic lift distribution, thereby reducing the induced drag.

One of the optimisation phase shapes looks somewhat different from its counterparts. This is the shape that was tried on iteration 102 and resulted in an increase in cost compared to the previous iteration. In the following iterations, it was not repeated.

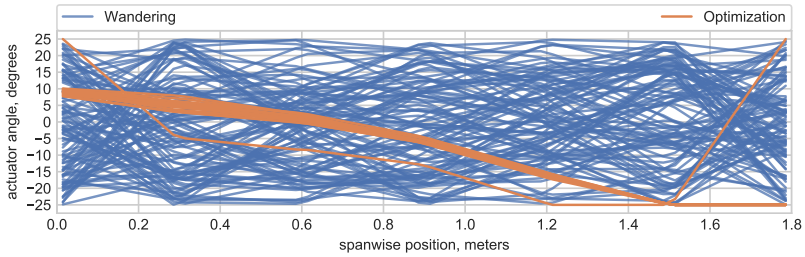


Figure II.2.18: Morphing shapes evaluated on the system in the wandering and optimisation phases.

II.2.4.2. Various Target Lift Coefficients

In order to investigate the ability of the online shape optimisation algorithm to find optimal inputs for different target lift coefficients without repeated exploring, the optimisation phase was extended to include two repeated series of steps and a window of gradual changes in the target lift coefficient as depicted in Fig. II.2.19. The quality of the solutions actuated on the system was also evaluated by comparing their lift-to-drag ratios to those of the wing jig shape.

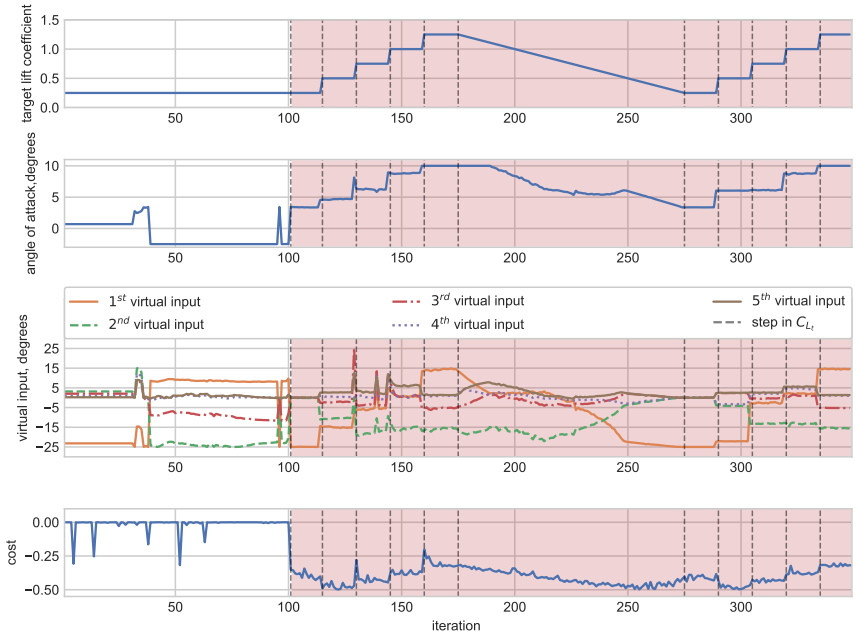


Figure II.2.19: Optimal inputs computed during the wandering and optimisation (red background) phases.

From 100 to 160, the target lift coefficient is increased by 0.25 every 15 iterations. The steps in the target lift coefficient are marked with dashed vertical lines. As a direct result

of the steps in the target lift coefficient, steps in the computed optimal angle of attack and virtual inputs can be seen at the corresponding iterations. For the duration of the steps, the optimal inputs are stable. The cost associated with the corresponding system outputs is also stable, although it is noisier due to the added measurement noise.

Between iterations 175 and 275, the target lift coefficient decreases from 1.25 to 0.25 in steps of 0.01. As expected, the optimal angle of attack and mean camber of the optimal shape decrease as the target lift coefficient decreases. The first virtual input, which contributes a constant amount of camber morphing along the wingspan, decreases until it nears the negative actuator limit of -25 degrees between iterations 175 and 248. Meanwhile, the second virtual input, representing a linear increase in spanwise camber morphing, becomes less harmful. Here the optimiser increases the negative u_2 input because the lower u_1 input leaves less room for spanwise lift reduction before the actuators at the wingtip hit their maximum negative deflections. Between iterations 248 and 275, virtual inputs u_2 through u_5 are decreased to zero so that u_1 can move all the way to the -25 degree actuator limit. In other words, for the target lift coefficient of 0.25, the optimiser sacrifices the increased lift induction efficiency of a more elliptical spanwise lift distribution for an overall less cambered airfoil. This makes sense since the airfoil is already relatively highly cambered, which is more efficient for producing higher lift coefficients.

After iteration 275, the same steps in the target lift coefficient are repeated. The optimal inputs are almost the same between the runs, with the exception of $C_{L_t} = 0.50$ during iterations 290-305. Even though the inputs are different in this case, the costs are very similar. The average cost during iterations 115-130 is -0.475 with a standard deviation of 0.011, whereas the average cost during iterations 290-305 is -0.481 with a standard deviation of 0.018. Hence, on average, the performance of the inputs evaluated during iterations 290-305 was slightly more desirable than those evaluated during iterations 115-130. Nevertheless, this again highlights the importance of accurate lift and drag estimations. Any combination of inputs can only be determined to be more efficient as long as the difference is measurable. In simulations without simulated measurement noise, the revisited target lift coefficients yielded the same inputs.

The lift coefficients and lift-to-drag ratios measured during the wandering and optimisation phases are shown together with those of the jig shape in Fig. II.2.20. As shown in Fig. II.2.20, the quasi-random shapes from the wandering phase, shown in blue, produce lower lift-to-drag ratios than the jig shape, shown in green, in roughly 80 % of the cases. Many possible shape variations exist that are aerodynamically inefficient, whereas only a smaller subset of shapes yield better aerodynamic performance. By chance, some random inputs perform comparably or even better than the jig shape.

Except for only two data points, the optimisation points, shown in orange, all outperform the jig shape in terms of aerodynamic efficiency. Although, for those two data points, the aerodynamic model output without simulated measurement noise does outperform the jig shape. Another effect of the measurement noise can be observed in the decreasing spread of the optimisation point cloud with increasing lift coefficients. Naturally, the lift-to-drag ratio becomes less sensitive to measurement noise as the lift and drag coefficients become larger. The clustering of optimisation points at the target lift coefficients that were repeated for multiple iterations indicates that the optimiser can achieve the target lift coefficient very closely whilst also outperforming the jig shape.

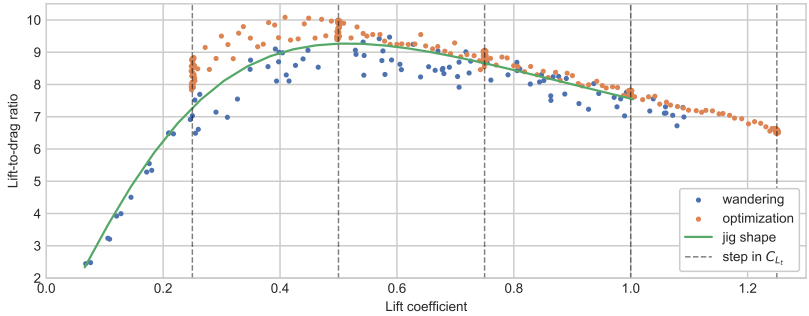


Figure II.2.20: Performance comparison of the jig shape and the online optimisation shapes.

An overview of the improvements in aerodynamic performance at various target lift coefficients is shown in Tab. II.2.2. As discussed, the relatively highly cambered airfoil is naturally efficient at inducing higher lift coefficients. This is why the highest performance increases from active wing morphing are observed for low lift coefficients (0.25-0.50). At $C_L = 0.25$ the lift-to-drag ratio is increased with approximately 14.6 %. At higher target lift coefficients, less increase in aerodynamic efficiency can be gained from changing the average amount of camber. At $C_L = 1.00$ the lift-to-drag ratio is increased with approximately 2.5 %. Due to the rectangular planform and absence of twist in the jig shape, reshaping the spanwise lift distribution closer to an elliptical distribution yields an aerodynamic performance increase at all target lift coefficients.

Table II.2.2: Efficiency improvements of the optimised wing shapes compared to the jig shape.

C_{L_t} [-]	C_D [-]	$\frac{L}{D}$ [-]	$\frac{L}{D}$ increase [%]	C_D reduction [%]
0.25	0.02995	8.35	14.6	12.8
0.50	0.05108	9.79	5.6	5.3
0.75	0.08420	8.91	2.9	2.8
1.00	0.12906	7.75	2.5	2.4

II.2.5. Conclusions and Recommendations

This Chapter presents a novel online learning-based black-box approach to active morphing wing shape optimisation. Its objective is to maximise the steady-state lift-to-drag ratio for a given target lift coefficient using lift and drag measurements. The presented method integrates an online-trained Artificial Neural Network (ANN) onboard model with an evolutionary optimisation algorithm. This optimisation strategy was tested in simulation on a seamless camber morphing wing model, and its performance was compared to the performance of the wing jig shape. Before optimising, the algorithm was allowed to explore the optimisation space with pseudo-random inputs in the wandering phase. Subsequently, in the optimisation phase, the optimiser used the onboard model to find the optimal wing

shape and angle of attack to achieve the target lift coefficient on the surrogate wing model.

During the wandering phase, the radial basis function neural networks could sufficiently learn the mapping between the angle of attack, wing shape, and the resulting aerodynamic forces to facilitate the optimiser to find wing shapes that outperformed the jig shape in terms of aerodynamic efficiency. Due to the global character of the neural network onboard model used, the presented optimisation strategy was able to find wing shape and angle of attack combinations with lift-to-drag ratio increases of up to 14.6 % for a wide range of target lift coefficients without requiring further exploration.

In the present case, the input space of the onboard model is comprised only of the wing shape and angle of attack. In actuality, mapping these parameters to the lift and drag coefficients is also influenced by the Reynolds number and Mach number. Nevertheless, due to the black-box nature of the neural network onboard model, future studies could, with little effort, incorporate the Reynolds and Mach numbers as additional inputs to expand its scope to the entire flight envelope of any camber morphing platform.

Finally, the current optimisation structure allows the implementation and development of other optimisers which can be integrated with the same black-box data structure in the inner loop. Furthermore, using the fast inner loop parallel query can be implemented of either CMA-ES or other optimiser to evaluate candidate solutions for the given target lift coefficient.

References

[1] O. Ruland, T. Mkhoyan, R. De Breuker, and X. Wang, "Black-box Online Aerodynamic Performance Optimization for a Seamless Wing with Distributed Morphing," *AIAA SciTech Forum 2022; In Review at Journal of Guidance, Control, and Dynamics*, p. 1840, 2022.

[2] D. Li, S. Zhao, A. Da Ronch, J. Xiang, J. Drofelnik, Y. Li, L. Zhang, Y. Wu, M. Kintscher, H. P. Monner, A. Rudenko, S. Guo, W. Yin, J. Kirn, S. Storm, and R. De Breuker, "A review of modelling and analysis of morphing wings," *Progress in Aerospace Sciences*, vol. 100, no. June, pp. 46–62, 2018.

[3] M. Wu, T. Xiao, H. Ang, and H. Li, "Optimal flight planning for a Z-shaped morphing-wing solar-powered unmanned aerial vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 2, pp. 497–505, 2018.

[4] H. R. Jex and F. E. C. Culick, "Flight Control Dynamics of the 1903 Wright Flyer," pp. 534–548, Aug. 1985.

[5] J. Hubbard Jr, "Dynamic shape control of a morphing airfoil using spatially distributed transducers," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 3, pp. 612–616, 2006.

[6] N. Nguyen, S. Lebofsky, E. Ting, U. Kaul, D. Chaparro, and J. Urnes, "Development of Variable Camber Continuous Trailing Edge Flap for Performance Adaptive Aeroelastic Wing," pp. 2015–01–2565, Sept. 2015.

[7] Y. Ferrier, N. Nguyen, and E. Ting, "Real-time adaptive least-squares drag minimization for performance adaptive aeroelastic wing," in *34th AIAA Applied Aerodynamics Conference*, (Washington, D.C.), American Institute of Aeronautics and Astronautics Inc, AIAA, 2016.

[8] N. Nguyen, N. B. Cramer, K. E. Hashemi, E. Ting, M. Drew, R. Wise, J. Boskovic, N. Precup, T. Mundt, and E. Livne, "Real-Time Adaptive Drag Minimization Wind Tunnel Investigation of a Flexible Wing with Variable Camber Continuous Trailing Edge Flap System," in *AIAA Aviation 2019 Forum*, (Dallas, Texas), American Institute of Aeronautics and Astronautics (AIAA), jun 2019.

[9] N. Nguyen and J. Xiong, "Real-time drag optimization of aspect ratio 13.5 common research model with distributed flap system," in *AIAA Scitech 2021 Forum*, (virtual event), American Institute of Aeronautics and Astronautics (AIAA), January 2021.

[10] J. F. Horn, E. M. Schmidt, B. R. Geiger, and M. P. DeAngelo, "Neural network-based trajectory optimization for unmanned aerial vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 548–562, 2012.

[11] A. Huang, Y. Luo, and H. Li, "Global Optimization of Multiple-Spacecraft Rendezvous Mission via Decomposition and Dynamics-Guide Evolution Approach," *Journal of Guidance, Control, and Dynamics*, pp. 1–8, 2021.

[12] J. Igarashi and D. B. Spencer, "Optimal continuous thrust orbit transfer using evolutionary algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 3, pp. 547–549, 2005.

[13] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies.," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[14] C. S. K. Dash, A. K. Behera, S. Dehuri, and S.-B. Cho, "Radial basis function neural networks: a topical state-of-the-art survey," *Open Computer Science*, vol. 6, no. 1, pp. 33–63, 2016.

[15] P. Shankar, R. K. Yedavalli, and J. J. Burken, "Self-organizing radial basis function networks for adaptive flight control," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 3, pp. 783–794, 2011.

[16] T. Mkhoyan, N. R. Thakrar, R. De Breuker, and J. Sodja, "Design and Development of a Seamless Smart Morphing Wing Using Distributed Trailing Edge Camber Morphing for Active Control," in *AIAA Scitech 2021 Forum*, (virtual event), p. 0477, American Institute of Aeronautics and Astronautics (AIAA), 2021.

[17] N. Werter, J. Sodja, G. Spirlet, and R. De Breuker, "Design and Experiments of a Warp Induced Camber and Twist Morphing Leading and Trailing Edge Device," in *24th AIAA/AHS Adaptive Structures Conference*, (San Diego, California, USA), American Institute of Aeronautics and Astronautics, Jan. 2016.

[18] X. Wang, T. Mkhoyan, I. Mkhoyan, and R. De Breuker, "Seamless Active Morphing Wing Simultaneous Gust and Maneuver Load Alleviation," *Journal of Guidance, Control, and Dynamics*, vol. 44, pp. 1649–1662, sep 2021.

[19] M. M. Gomroki, F. Topputo, F. Bernelli-Zazzera, and O. Tekinalp, "Solving Constrained Optimal Control Problems Using State-Dependent Factorization and Chebyshev Polynomials," *Journal of Guidance, Control, and Dynamics*, vol. 41, pp. 618–631, mar 2018.

[20] D. Löbl, F. Holzapfel, M. Weiss, and T. Shima, "Cooperative Docking Guidance and Control with Application to Civil Autonomous Aerial Refueling," *Journal of Guidance, Control, and Dynamics*, pp. 1–11, 2021.

[21] P. Sharpe, "Aerosandbox 2.2.11." PyPi, Sep. 30, 2020. [Online].

[22] J. S. Richter, J. B. Woodring, S. E. Fox, and R. K. Agarwal, "Performance study of a tapered flying wing with bell-shaped lift distribution," in *AIAA Scitech 2021 Forum*, (virtual event), American Institute of Aeronautics and Astronautics (AIAA), 2021.

[23] G. Molinari, A. F. Arrieta, and P. Ermanni, "Planform, aero-structural and flight control optimization for tailless morphing aircraft," *Journal of Intelligent Material Systems and Structures*, vol. 29, pp. 3847–3872, Dec. 2018.

[24] D. Keidel, G. Molinari, and P. Ermanni, "Aero-structural optimization and analysis of a camber-morphing flying wing: Structural and wind tunnel testing," *Journal of Intelligent Material Systems and Structures*, vol. 30, no. 6, pp. 908–923, 2019.

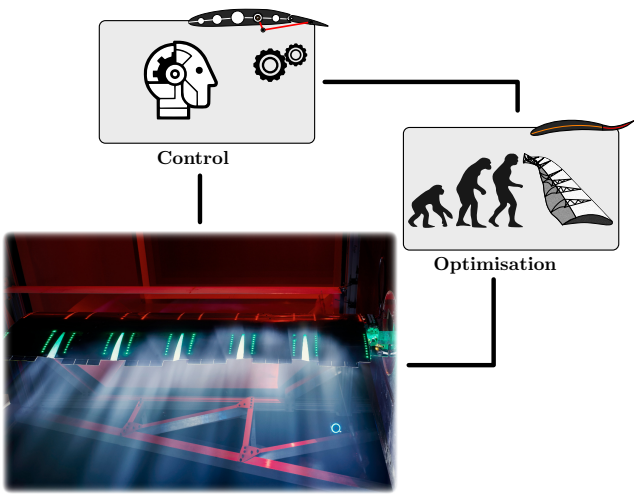
[25] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright, "Randomized smoothing for (parallel) stochastic optimization," in *Proceedings of the IEEE Conference on Decision and Control*, (Maui, Hawaii), pp. 5442–5444, Institute of Electrical and Electronics Engineers (IEEE), 2012.

[26] M. McCloskey and N. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of Learning and Motivation - Advances in Research and Theory*, vol. 24, pp. 109–165, Jan. 1989.

[27] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, (New Orleans, Louisiana), 2018.

III

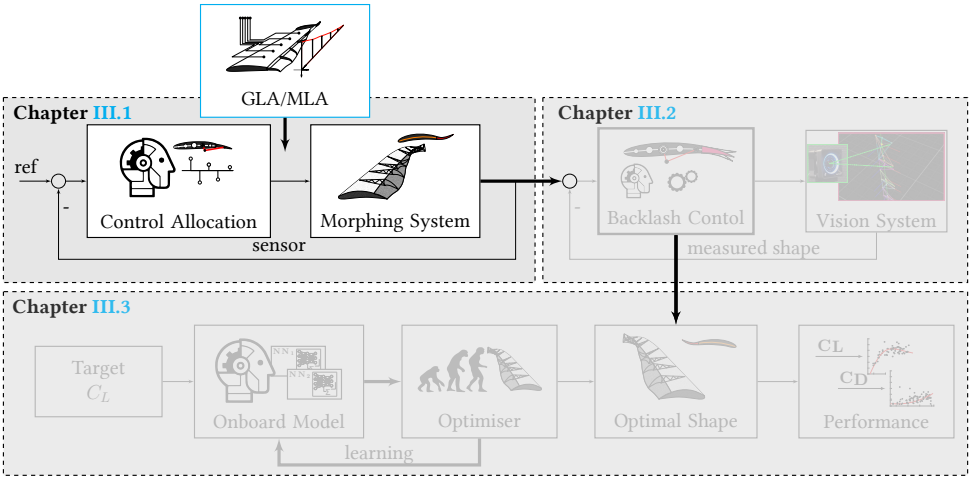
Smart Control & Integration



III.1

Simultaneous Gust and Manoeuvre Load Alleviation

In Part II, the development and the realisation of the seamless active morphing wing concept were presented. This Chapter presents the design and wind tunnel testing of a simultaneous gust and manoeuvre load alleviation control law for achieving multiple desired objectives of the smart morphing wing. The sensor-based incremental control allocation approach using Incremental Nonlinear Dynamic Inversion and Quadratic Programming (INDI-QP) is implemented for the system to satisfy actuator position constraints, rate constraints, and relative position constraints. The approach is augmented with the Virtual shape functions (denoted as INDI-QP-V) to ensure the smoothness of the morphing wing [1].



This Chapter is based on the following journal paper:
X. Wang, T. Mkhoyan, I. Mkhoyan, and R. De Breuker, "Seamless Active Morphing Wing Simultaneous Gust and Maneuver Load Alleviation," *Journal of Guidance, Control, and Dynamics*, pp. 1649–1662, 2021

III.1

This Chapter deals with the simultaneous gust and manoeuvre load alleviation objective of a seamless active morphing wing. The incremental nonlinear dynamic inversion with quadratic programming control allocation and virtual shape functions (denoted as INDI-QP-V) is proposed to fulfil this goal. The designed control allocator provides an optimal solution while satisfying actuator position constraints, rate constraints, and relative position constraints. Virtual shape functions ensure the smoothness of the morphing wing at every moment. In the presence of model uncertainties, external disturbances, and control allocation errors, the closed-loop stability is guaranteed in the Lyapunov sense. Wind tunnel tests demonstrate that INDI-QP-V can make the seamless wing morph actively to resist "1-cos" gusts and modify the spanwise lift distribution at the same time. The wing root shear force and bending moment have been alleviated by more than 44 % despite unexpected actuator fault and nonlinear backlash. Moreover, during the experiment, all the input constraints were satisfied, the wing shape was smooth all the time, and the control law was executed in real-time. Furthermore, as compared to the linear quadratic Gaussian (LQG) control, the hardware implementation of INDI-QP-V is easier; the robust performance of INDI-QP-V is also superior.

III.1.1. Introduction

The advancements in aerospace engineering, paired with continuing desire to develop more fuel-efficient aircraft, lead to increasingly flexible aircraft designs. Generally, the flexibility is considered as a side effect of the lighter aircraft design and needs to be adequately accounted for to prevent undesired aerodynamics-structure couplings and ensure the optimised aerodynamic shape. While the flexibility can be accounted for with either passively tailored structural design or active control mechanisms, a fixed-wing shape - generally optimised for the cruise condition - cannot be fully optimised throughout the flight envelope due to conflicting requirements [2]. A more natural approach is to *utilise* the flexibility and actively change the shape by in-flight morphing. This allows the wing to continuously adapt to the most optimal shape when transitioning from one flight phase to the other. Secondly, as compared to the conventional discrete trailing-edge surfaces, smooth morphing can execute flight control and load alleviation commands with reduced noise and drag. The combination of these two aspects can contribute to a more efficient flight routine and a reduced structural weight, thereby improving flight sustainability.

Many research efforts have been devoted to morphing, including piezoelectricity, shape memory alloys materials, a compliant actuation mechanism, etc. [3]. Among all these research aspects, actuation force reduction is one of the bottlenecks of morphing realisations. In [4], the actuation force is reduced by a compliant skin mechanism, and a combination of conventional and piezoelectric actuation. However, this approach results in significant manufacturing challenges and complexities. Other concepts, such as the Fish Bone Active Camber (FishBAC) [5] and the Mission Adaptive Digital Composite Aerostructure Technologies (MADCAT) [6] demonstrate morphing with ultralight structures. However, the majority of the wing volume is consumed for morphing mechanisms, leaving limited room for other components.

Overviewing the state of the art, the key shortcomings of existing morphing techniques include 1) restricted morphing motions; 2) manufacturability and scalability complexities; 3) compromised internal wing volume; 4) inadmissibility for distributed morphing control along the wing span. To overcome these shortcomings, a distributed seamless active morphing wing concept is proposed in [7]. As shown in Fig. III.1.1, this morphing wing named SmartX-Alpha is based on the Translation Induced Camber (TRIC) con-

cept [7], which means a cut is introduced to allow the bottom skin to slide in chordwise and transverse directions. By altering the actuation directions, a pair of actuators can introduce pure camber morphing or warp-induced spanwise twist morphing. To ensure seamlessness, the adjacent TRIC modules are connected with elastomeric skin, whose stiffness is designed considering the aerodynamic shape holding and the actuation loads. The distributed modular composite design with released internal structural stresses makes it relatively easy to scale up the SmartX-Alpha wing to a full-scale flight aircraft. The control algorithms proposed in this Chapter will be applied to the SmartX-Alpha morphing wing[1].

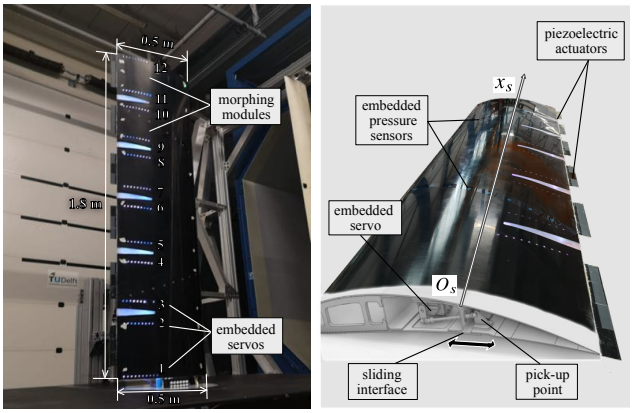


Figure III.1.1: The seamless active morphing wing SmartX-Alpha with the TRIC morphing mechanism.

Manoeuvre Load Alleviation (MLA) and Gust Load Alleviation (GLA) are two important objectives in aircraft control. Conventional MLA relies on some pre-designed control logic, e.g., when the measured load exceeds a pre-defined threshold, the pre-selected wing control surfaces are triggered to deflect and hold for a certain time period [8]. However, this conventional approach is not efficient and also requires intensive tuning efforts. By contrast, Pereira et al. [9] use a linear Model Predictive Controller (MPC) and a Linear Quadratic Regulator (LQR) to satisfy the load constraints at various critical stations. In a study by Haghighat et al. [10], the nonlinear flexible aircraft model is linearised successively, and then MPC controllers are designed at every linearisation point. A Linear Quadratic Gaussian (LQG) control is designed for a SensorCraft vehicle GLA problem in [11]. Another experimental study in the Open Jet Facility (OJF) investigated the potential of load reduction with smart rotors using H_∞ loop shaping and Feed Forward control [12]. Besides, a wind tunnel experiment for alleviating the gust loads of a flexible wing with piezoelectric control is presented in [13]. The piezoelectric patches are actuated by a Proportional-Integral-Derivative (PID) controller using wing-tip linear acceleration measurements. In [14], an aeroelastic morphing vehicle is controlled using Linear Parameter-Varying (LPV) and pole placement techniques.

In the study by Nguyen et al. [15], a distributed control surface layout named Variable Continuous Camber Trailing Edge Morphing (VCCTEF) is used to achieve multi-objective

[1]The project video can be found via https://www.youtube.com/watch?v=SdagIiYRWyA&t=319s

flight control and manoeuvre load alleviation. Two cost functions are used in the LQG control, one for rigid-body command tracking, and another for elastic mode suppression and wing root bending moment minimisation [15]. Simulation results in [15] show that the pitch rate tracking performance is degraded by the MLA function. However, for an aircraft with distributed wing control surfaces, it is physically realistic to simultaneously achieve the necessary loads for command tracking, while alleviating the excessive loads caused by manoeuvres and gusts. The LQG controller in [15] is augmented with an adaptive GLA function in [16]. Based on the differences between the measured and model-predicted accelerations, the gust components on rigid-body and elastic dynamics are estimated online using a gradient adaptive law. However, because the number of accelerometer outputs is less than the number of gust load elements, the gust estimation is only in a least-squares sense. Moreover, the resulting estimations are not purely gusts but also contain model uncertainties. Furthermore, as commented in [17], atmospheric disturbances have high-frequency components, which would require prohibitively high learning rates of adaptation laws.

Different from the above-mentioned linear model-based control techniques, an Incremental Non-linear Dynamic Inversion (INDI) control law is proposed in [18] for alleviating the gust loads of a flexible aircraft. In contrast to the linear control methods, INDI does not need the tedious gain-scheduling process when applied to nonlinear dynamic systems. In comparison to other model-based nonlinear control methods such as feedback linearisation [19] and backstepping [20], INDI has less model dependency, which simplifies its implementation process. Although its model dependency is reduced, the robustness of INDI is actually enhanced by exploiting the sensor measurements. Experimental and simulation results have demonstrated the robustness of INDI to model uncertainties [21], gust disturbances [18], actuator faults [22], and structural damage [23]. As opposed to adaptive control methods, INDI does not need the uncertainty parameterisation process nor the assumption of slowly time-varying uncertain parameters. Additionally, its computational load is also lower than the adaptive control methods.

The flexible aircraft configuration used in [18] only has one aileron on each wing. Consequently, within the INDI control loop, trade-offs among different virtual control components have to be made. Besides, input constraints are not considered in [18]. For the SmartX-Alpha morphing wing with distributed actuation, wing load alleviation becomes an over-actuated problem, where control allocation is needed. Moreover, it is crucial to constrain the relative deflections between adjacent morphing modules to avoid over-stretching the elastomer. In the literature, an Incremental Non-linear Control Allocation (INCA) method has been proposed for a tailless aircraft with Innovative Control Effectors (ICE) [24].

While this approach provided effective control allocation for a highly manoeuvrable nonlinear aircraft model, the relative position constraints were not considered. In order to enforce smoothness of the lift distribution and allocation for a distributed morphing concept, relative position constraints and consequent smooth actuator allocation are critical. Furthermore, the closed-loop stability in the presence of model uncertainties, disturbances, and possible control allocation must be addressed.

The contributions of this Chapter are: (i) INDI method with Quadratic Programming Control Allocation (INDI-QP) considering actuator *relative* position constraints, position

constraints, and rate constraints; (ii) INDI-QP method augmented with virtual shape functions (denoted as INDI-QP-V), which can ensure the smoothness of a morphing wing at every moment. (iii) implementation, evaluation and wind tunnel testing of INDI-QP-V on a simultaneous gust and manoeuvre load alleviation problem of a seamless active morphing wing; (iv) robust load alleviation performance comparisons between INDI-QP-V and LQG control in the presence of actuator fault and nonlinear backlash. The rest of this Chapter is structured as follows. Section III.1.2 derives the control algorithms. The experimental results are presented in Sec. III.1.3. The proposed INDI-QP-V is compared to LQG in Sec. III.1.4. Main conclusions are drawn in Sec. III.1.5.

III.1.2. Incremental Control Design

This section describes the control design for the seamless morphing wing.

III.1.2.1. Incremental Control Theory

Considering the following nonlinear multi-input multi-output system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d}(t), \quad \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \tag{III.1.1}$$

where $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^p$ are smooth vector fields. $\boldsymbol{G}$ is a smooth function mapping $\mathbb{R}^n \to \mathbb{R}^{n\times m}$, whose columns are smooth vector fields. $\boldsymbol{d}(t) \in \mathbb{R}^n$ represents the external disturbance vector. Assume $\|\boldsymbol{d}(t)\|_2 \leq \bar{d}$. $\boldsymbol{y} \in \mathbb{R}^p$ in Eq. (III.1.1) denotes the controlled output vector, which can be a function of any subset of the physical measurable outputs. This Chapter considers the case where $p \leq m$. Define the vector relative degree [25] of the system as $\boldsymbol{\rho} = [\rho_1, \rho_2, ..., \rho_p]^{\mathrm{T}}$, which satisfies $\rho = \|\boldsymbol{\rho}\|_1 = \sum_{i=1}^{p} \rho_i \leq n$, then by differentiating the output vector $\boldsymbol{y}$, the input-output mapping is given as

$$\boldsymbol{y}^{(\boldsymbol{\rho})} = \boldsymbol{\alpha}(\boldsymbol{x}) + \boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d}_y \tag{III.1.2}$$

In Eq. (III.1.2), $\boldsymbol{\alpha}(\boldsymbol{x}) = [\mathcal{L}_f^{\rho_1} h_1, \mathcal{L}_f^{\rho_2} h_2, ..., \mathcal{L}_f^{\rho_p} h_p]^{\mathrm{T}}$, $\boldsymbol{\mathcal{B}}(\boldsymbol{x}) \in \mathbb{R}^{p\times m}$, $\mathcal{B}_{ij} = \mathcal{L}_{g_j}\mathcal{L}_f^{\rho_i - 1} h_i$, where $\mathcal{L}_f^{\rho_i} h_i$, $\mathcal{L}_{g_j}\mathcal{L}_f^{\rho_i - 1} h_i$ are the corresponding Lie derivatives [26]. When $\rho_i = 1$ for all $i = 1, 2, ..., p$, $\boldsymbol{d}_y = [\mathcal{L}_d h_1, \mathcal{L}_d h_2, ..., \mathcal{L}_d h_p]^{\mathrm{T}}$. For more general cases where $\rho_i > 1$, $\boldsymbol{d}_y$ also contains the cross-coupling terms of $\mathcal{L}_d h_i$ and $\mathcal{L}_f h_i$. If $\rho = n$, then the system given by Eq. (III.1.1) is full-state feedback linearisable. Otherwise, there exist $n - \rho$ internal dynamics. Denote the sampling interval as Δt, the incremental dynamic equation is derived by taking the first-order Taylor series expansion of Eq. (III.1.2) around the condition at $t - \Delta t$ (denoted by the subscript 0) as:

$$\boldsymbol{y}^{(\boldsymbol{\rho})} = \boldsymbol{y}_0^{(\boldsymbol{\rho})} + \left.\frac{\partial[\boldsymbol{\alpha}(\boldsymbol{x}) + \boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u}]}{\partial \boldsymbol{x}}\right|_0 \Delta\boldsymbol{x} + \boldsymbol{\mathcal{B}}(\boldsymbol{x}_0)\Delta\boldsymbol{u} + \Delta\boldsymbol{d}_y + \boldsymbol{R}_1 \tag{III.1.3}$$

in which $\Delta\boldsymbol{x}$, $\Delta\boldsymbol{u}$, and $\Delta\boldsymbol{d}_y$ respectively represents the state, control, and disturbance increments in one Δt. $\boldsymbol{R}_1$ is the expansion remainder. Consider the output tracking problem and denote the reference as $\boldsymbol{y}_r(t) = [y_{r_1}(t), y_{r_2}(t), ..., y_{r_p}(t)]^{\mathrm{T}}$. Assume that $y_{r_i}(t)$, $i = 1, 2, ..., p$, and its derivatives up to $y_{r_i}^{(\rho_i)}(t)$ are bounded for all t and each $y_{r_i}^{(\rho_i)}(t)$ is continuous. Then the tracking error vector yields $\boldsymbol{e} = \boldsymbol{\xi} - \boldsymbol{\mathcal{R}}$, $\boldsymbol{\mathcal{R}} = [\boldsymbol{\mathcal{R}}_1^{\mathrm{T}}, \boldsymbol{\mathcal{R}}_2^{\mathrm{T}}, ..., \boldsymbol{\mathcal{R}}_p^{\mathrm{T}}]^{\mathrm{T}}$,

III.1

$\boldsymbol{\mathcal{R}}_i = [y_{r_i}, y_{r_i}^{(1)}, ..., y_{r_i}^{(\rho_i-1)}]^{\mathrm{T}}$. Assume $\|\boldsymbol{\mathcal{R}}\|_2 \leq \bar{\mathcal{R}}$. To stabilise the error dynamics, the control increment is designed to satisfy the following equation:

$$\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\Delta\boldsymbol{u}_{\text{indi}} = \boldsymbol{\nu}_c - \boldsymbol{y}_0^{(\boldsymbol{\rho})}, \quad \boldsymbol{\nu}_c = \boldsymbol{y}_r^{(\boldsymbol{\rho})} - \boldsymbol{K}\boldsymbol{e} \tag{III.1.4}$$

where $\bar{\boldsymbol{\mathcal{B}}}$ is an estimation of $\boldsymbol{\mathcal{B}}$. The gain matrix $\boldsymbol{K} = \text{diag}\{\boldsymbol{K}_i\}$, $i = 1, 2, ..., p$, and $\boldsymbol{K}_i = [K_{i,0}, K_{i,1}, ..., K_{i,\rho_i-1}]$. $\boldsymbol{y}_0^{(\boldsymbol{\rho})}$ is directly measured or estimated. The total control command for actuator is $\boldsymbol{u}_{\text{indi}} = \boldsymbol{u}_{\text{indi},0} + \Delta\boldsymbol{u}_{\text{indi}}$. Assume the row rank of $\bar{\boldsymbol{\mathcal{B}}}$ equals p. If the column rank of $\bar{\boldsymbol{\mathcal{B}}}$ also equals p, then there exists a unique $\Delta\boldsymbol{u}_{\text{indi}}$ satisfying Eq. (III.1.4). If the column rank of $\bar{\boldsymbol{\mathcal{B}}}$ is less than p, then the system is under-actuated and Eq. (III.1.4) cannot be satisfied. If the column rank of $\bar{\boldsymbol{\mathcal{B}}}$ is larger than p, then solving $\Delta\boldsymbol{u}_{\text{indi}}$ from Eq. (III.1.4) is a control allocation problem. Without considering the input constraints, there are infinite $\Delta\boldsymbol{u}_{\text{indi}}$ that satisfy Eq. (III.1.4). However, when some dimensions of $\boldsymbol{u}_{\text{indi}}$ get saturated, it is possible that $\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\Delta\boldsymbol{u}_{\text{indi}} - (\boldsymbol{\nu}_c - \boldsymbol{y}_0^{(\boldsymbol{\rho})}) \neq \boldsymbol{0}$ *even though* the column rank of $\bar{\boldsymbol{\mathcal{B}}}$ is higher than p. To make the theoretical analyses more general, Eq. (III.1.4) is generalised to $\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\Delta\boldsymbol{u}_{\text{indi}} = \boldsymbol{\nu}_c - \boldsymbol{y}_0^{(\boldsymbol{\rho})} + \boldsymbol{\varepsilon}_{\text{ca}}$, with $\boldsymbol{\varepsilon}_{\text{ca}}$ as the possible control allocation error. Considering the internal dynamics, the resulting closed-loop dynamics are:

$$\begin{aligned}
\dot{\boldsymbol{\eta}} &= \boldsymbol{f}_\eta(\boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{d}) = \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}}(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{d}(t))\Big|_{\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{z})} \\
\dot{\boldsymbol{e}} &= (\boldsymbol{A}_c - \boldsymbol{B}_c\boldsymbol{K})\boldsymbol{e} + \boldsymbol{B}_c[\boldsymbol{\delta}(\boldsymbol{x}, \Delta t) + (\boldsymbol{\mathcal{B}}(\boldsymbol{x}_0) - \bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0))\Delta\boldsymbol{u}_{\text{indi}} + \boldsymbol{\varepsilon}_{\text{ca}} + \Delta\boldsymbol{d}_y] \\
&\triangleq (\boldsymbol{A}_c - \boldsymbol{B}_c\boldsymbol{K})\boldsymbol{e} + \boldsymbol{B}_c\boldsymbol{\varepsilon}_{\text{indi}}
\end{aligned} \tag{III.1.5}$$

whereby $\boldsymbol{\eta}$ represents the internal state vector, and $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x}) = [\boldsymbol{\eta}^{\mathrm{T}}, \boldsymbol{\xi}^{\mathrm{T}}]^{\mathrm{T}}$, with $\boldsymbol{\eta} = \boldsymbol{\phi}(\boldsymbol{x})$, $\boldsymbol{\xi} = [\boldsymbol{\xi}_1^{\mathrm{T}}, ..., \boldsymbol{\xi}_p^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{\xi}_i = [h_i(\boldsymbol{x}), ..., \mathcal{L}_f^{\rho_i-1}h_i(\boldsymbol{x})]^{\mathrm{T}}$, $i = 1, 2, ..., p$ is a diffeomorphism. $\boldsymbol{\delta}(\boldsymbol{x}, \Delta t)$ is the closed-loop value of the variations and expansion reminder: $\boldsymbol{\delta}(\boldsymbol{x}, \Delta t) = \left[\frac{\partial[\boldsymbol{\alpha}(\boldsymbol{x})+\boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u}]}{\partial \boldsymbol{x}}\Big|_0\Delta\boldsymbol{x} + \boldsymbol{R}_1\right]\Big|_{\boldsymbol{u}=\boldsymbol{u}_{\text{indi}}}$. $\boldsymbol{A}_c = \text{diag}\{\boldsymbol{A}_0^i\}$, $\boldsymbol{B}_c = \text{diag}\{\boldsymbol{B}_0^i\}$, $\boldsymbol{C}_c = \text{diag}\{\boldsymbol{C}_0^i\}$, $i = 1, 2, ..., p$, and $(\boldsymbol{A}_0^i, \boldsymbol{B}_0^i, \boldsymbol{C}_0^i)$ is a canonical form representation of a chain of ρ_i integrators. The gain matrix $\boldsymbol{K}$ is designed such that $\boldsymbol{A}_c - \boldsymbol{B}_c\boldsymbol{K}$ is Hurwitz.

In contrast to the model-based feedback linearisation, the INDI is a sensor-based control strategy [27]. By exploiting the sensor measurements, the only model information needed by INDI is the estimated control effectiveness matrix $\bar{\boldsymbol{\mathcal{B}}}$, which simplifies the implementation process. Moreover, the residual perturbation in the closed-loop system is also reduced, which enhances the control robustness against model uncertainties, external disturbances, and sudden faults [23].

Remark 1. A stability analysis for INDI that simultaneously considers control allocation errors, internal dynamics, model uncertainties, and external disturbances has not been addressed in the literature. In view of this, the following two theorems are proposed in this Chapter:

Theorem 1. *If $\|\boldsymbol{\varepsilon}_{indi}\|_2 \leq \bar{\varepsilon}$ is satisfied for all $\boldsymbol{\xi} \in \mathbb{R}^\rho$, $\boldsymbol{f}_\eta(\boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{d})$ is continuously differentiable and globally Lipschitz in $(\boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{d})$, and the origin of $\dot{\boldsymbol{\eta}} = \boldsymbol{f}_\eta(\boldsymbol{\eta}, \boldsymbol{0}, \boldsymbol{0})$ is globally exponentially stable, then the tracking error $\boldsymbol{e}$ in Eq. (III.1.5) is globally ultimately bounded by a class $\mathcal{K}$ function of $\bar{\varepsilon}$, while the internal state $\boldsymbol{\eta}$ in Eq. (III.1.5) is globally ultimately bounded by a class $\mathcal{K}$ function of $\bar{\varepsilon}$, $\bar{\mathcal{R}}$, and $\bar{d}$.*

Proof: See Appendix.

Theorem 2. *If $\|\boldsymbol{\varepsilon}_{indi}\|_2 \leq \bar{\varepsilon}$ is satisfied for all $\boldsymbol{\xi} \in \mathbb{R}^{\rho}$, $\boldsymbol{f}_{\eta}(\boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{d})$ is continuously differentiable, and the origin of $\dot{\boldsymbol{\eta}} = \boldsymbol{f}_{\eta}(\boldsymbol{\eta}, \boldsymbol{0}, \boldsymbol{0})$ is exponentially stable, then there exists a neighborhood D_z of $\boldsymbol{z} = [\boldsymbol{0}^{\mathrm{T}}, \boldsymbol{\mathcal{R}}^{\mathrm{T}}]^{\mathrm{T}}$ and $\varepsilon^* > 0$, such that for every $\boldsymbol{z}(t=0) \in D_z$ and $\bar{\varepsilon} < \varepsilon^*$, the tracking error $\boldsymbol{e}$ in Eq. (III.1.5) is ultimately bounded by a class $\mathcal{K}$ function of $\bar{\varepsilon}$, while the internal state $\boldsymbol{\eta}$ in Eq. (III.1.5) is ultimately bounded by a class $\mathcal{K}$ function of $\bar{\varepsilon}$, $\bar{\mathcal{R}}$, and $\bar{d}$.*

Proof: See Appendix.

III.1.2.2. INCREMENTAL CONTROL ALLOCATION

This subsection will solve $\Delta \boldsymbol{u}_{\text{indi}}$ from Eq. (III.1.4), and discuss the corresponding boundedness conditions for $\boldsymbol{\varepsilon}_{\text{indi}}$ (Eq. (III.1.5)). The control allocation problem considers the case that the row rank of $\bar{\boldsymbol{\mathcal{B}}} \in \mathbb{R}^{p \times m}$ equals p, while its column rank is larger than p. Under this condition, Eq. (III.1.4) is satisfied by:

$$\Delta \boldsymbol{u}_{\text{indi}} = \bar{\boldsymbol{\mathcal{B}}}^{+}(\boldsymbol{x}_0)(\boldsymbol{\nu}_c - \boldsymbol{y}_0^{(\boldsymbol{\rho})}) + (\boldsymbol{I}_{m\times m} - \bar{\boldsymbol{\mathcal{B}}}^{+}(\boldsymbol{x}_0)\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0))\boldsymbol{w} \tag{III.1.6}$$

Here, $\bar{\boldsymbol{\mathcal{B}}}^{+} = \bar{\boldsymbol{\mathcal{B}}}^{T}(\bar{\boldsymbol{\mathcal{B}}}\bar{\boldsymbol{\mathcal{B}}}^{T})^{-1}$ is the Moore-Penrose inverse of $\bar{\boldsymbol{\mathcal{B}}}$. It is noteworthy that although $\bar{\boldsymbol{\mathcal{B}}}\bar{\boldsymbol{\mathcal{B}}}^{+} = \boldsymbol{I}_{p\times p}$, $\bar{\boldsymbol{\mathcal{B}}}^{+}\bar{\boldsymbol{\mathcal{B}}} \neq \boldsymbol{I}_{m\times m}$. Besides, $\boldsymbol{w}$ can be any vector in $\mathbb{R}^{m\times 1}$. Nevertheless, $\Delta \boldsymbol{u}_{\text{indi}}$ only has the smallest Euclidean norm when $\boldsymbol{w} = \boldsymbol{0}$. This least squares solution given by pseudo-inverse is:

$$\Delta \boldsymbol{u}_{\text{indi-pi}} = \bar{\boldsymbol{\mathcal{B}}}^{+}(\boldsymbol{x}_0)(\boldsymbol{\nu}_c - \boldsymbol{y}_0^{(\boldsymbol{\rho})}) \tag{III.1.7}$$

Theorem 3. *When the pseudo-inverse control allocation is used (as presented in Eq. (III.1.7)), if $\|\boldsymbol{I} - \boldsymbol{\mathcal{B}}(\boldsymbol{x}_0)\bar{\boldsymbol{\mathcal{B}}}^{+}(\boldsymbol{x}_0)\|_2 \leq \bar{b} < 1$, and if $\boldsymbol{\delta}(\boldsymbol{x}, \Delta t)$ and $\Delta \boldsymbol{d}_y$ are respectively bounded by $\bar{\delta}$ and $\overline{\Delta d}$, then under sufficiently high sampling frequency, $\boldsymbol{\varepsilon}_{indi}$ in Eq. (III.1.5) is ultimately bounded.*

Proof: See Appendix.

Theorem 3 presents that one of the sufficient conditions for the boundedness of $\boldsymbol{\varepsilon}_{\text{indi}}$ is a diagonally dominated $\boldsymbol{\mathcal{B}}(\boldsymbol{x}_0)\bar{\boldsymbol{\mathcal{B}}}^{+}(\boldsymbol{x}_0)$. If this condition is satisfied, then the influences of model mismatches can be automatically tolerated by the controller. Otherwise, online model identification and adaptation for $\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)$ can be needed.

Although pseudo-inverse can provide the least squares solution, the input constraints are not considered. The servo position constraints are formulated as $\boldsymbol{u}_{\min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max}$, and can be rewritten as a linear inequality $[\boldsymbol{I}_{m\times m}, -\boldsymbol{I}_{m\times m}]^{\mathrm{T}}(\Delta \boldsymbol{u} + \boldsymbol{u}_0) \leq [\boldsymbol{u}_{\max}^{\mathrm{T}}, -\boldsymbol{u}_{\min}^{\mathrm{T}}]^{\mathrm{T}}$. The servos also have rate limits, i.e., $\underline{\boldsymbol{u}}_{\text{rate}}\, \Delta t \leq \Delta \boldsymbol{u} \leq \bar{\boldsymbol{u}}_{\text{rate}}\, \Delta t$. Furthermore, to avoid the elastomer between the morphing modules being over-stretched, the relative command differences between adjacent servos also need to be constrained.

For $\boldsymbol{u} \in \mathbb{R}^m$, there are $m-1$ relative position constraints, denoted as $\bar{\boldsymbol{u}}_{\text{adj}} \in \mathbb{R}^{m-1}$, with $|u_{i+1} - u_i| \leq \bar{u}_{\text{adj},i}$, $i = 1, 2, \ldots, m-1$. The elements of $\bar{\boldsymbol{u}}_{\text{adj}}$ are not necessarily equal. For example, regarding two adjacent servos in the SmartX-Alpha, if it is an

elastomer between them, then the relative actuation limit is set as 10 deg to prevent overstretching. Otherwise, the relative limit is relaxed to 55 deg. The relative position constraints are formulated as the following inequality: $[\boldsymbol{C}, -\boldsymbol{C}]^{\mathrm{T}}(\Delta\boldsymbol{u}+\boldsymbol{u}_0) \leq [\bar{\boldsymbol{u}}_{\text{adj}}^{\mathrm{T}}, \bar{\boldsymbol{u}}_{\text{adj}}^{\mathrm{T}}]^{\mathrm{T}}$. $\boldsymbol{C} \in \mathbb{R}^{(m-1)\times m}$, with $C_{i,i}=1$, $C_{i,i+1}=-1$ for $i=1,2,...,m-1$. Besides, the rest elements of $\boldsymbol{C}$ are all equal to zero. Considering the servo position, rate, and relative position limits, the control increment vector $\Delta\boldsymbol{u}$ has to satisfy the following inequality:

$$\begin{bmatrix} \boldsymbol{I}_{m\times m} \\ -\boldsymbol{I}_{m\times m} \\ \boldsymbol{C} \\ -\boldsymbol{C} \\ \boldsymbol{I}_{m\times m} \\ -\boldsymbol{I}_{m\times m} \end{bmatrix} \Delta\boldsymbol{u} \leq \begin{pmatrix} \boldsymbol{u}_{\max}-\boldsymbol{u}_0 \\ -\underline{\boldsymbol{u}}_{\min}+\boldsymbol{u}_0 \\ \bar{\boldsymbol{u}}_{\text{adj}}-\boldsymbol{C}\boldsymbol{u}_0 \\ \bar{\boldsymbol{u}}_{\text{adj}}+\boldsymbol{C}\boldsymbol{u}_0 \\ \bar{\boldsymbol{u}}_{\text{rate}}\Delta t \\ -\underline{\boldsymbol{u}}_{\text{rate}}\Delta t \end{pmatrix}, \quad \text{denoted as} \quad \boldsymbol{A}_u\Delta\boldsymbol{u} \leq \boldsymbol{b}_u \tag{III.1.8}$$

Remark 2. Equation (III.1.8) presents the first work that converts the actuator position constraints, rate constraints, and *relative* position constraints into an integrated linear inequality matrix with respect to the incremental control vector $\Delta\boldsymbol{u}$.

From a theoretical point of view, the linear equality constraint in Eq. (III.1.4) has the highest priority. If both Eq. (III.1.4) and the inequality constraint in Eq. (III.1.8) can be satisfied, then the rest free space of $\Delta\boldsymbol{u}$ can be used to minimise the energy of $\boldsymbol{u}$. However, under some faulty conditions, the feasible region can become null if both the equality (Eq. (III.1.4)) and the inequality (Eq. (III.1.8)) constraints are imposed. Actually, it is more practical to satisfy the inequality first and then minimise the realisation error of the equality constraint. For example, consider an actuator fault condition where Eq. (III.1.4) and Eq. (III.1.8) cannot be simultaneously satisfied; it is more meaningful to realise Eq. (III.1.8) first and allow certain performance degradation, rather than enforcing Eq. (III.1.4) by violating Eq. (III.1.8). Therefore, the first cost function is formulated as $\mathcal{J}_1 = (1/2)(\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\Delta\boldsymbol{u}-\boldsymbol{\nu}_c+\boldsymbol{y}_0^{(\boldsymbol{\rho})})^{\mathrm{T}}\boldsymbol{W}_1(\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\Delta\boldsymbol{u}-\boldsymbol{\nu}_c+\boldsymbol{y}_0^{(\boldsymbol{\rho})})$, where $\boldsymbol{W}_1$ is a positive definite weighting matrix.

Apart from realising Eq. (III.1.4), the control allocator should make $\boldsymbol{u}$ close to its nominal value $\boldsymbol{u}_*$. A typical choice is $\boldsymbol{u}_* = \boldsymbol{0}$ for minimising the control energy. For a morphing wing, $\boldsymbol{u}_*$ can also be non-zero to achieve an optimised wing shape. Therefore, the second cost function is $\mathcal{J}_2 = (1/2)(\Delta\boldsymbol{u}+\boldsymbol{u}_0-\boldsymbol{u}_*)^{\mathrm{T}}\boldsymbol{W}_2(\Delta\boldsymbol{u}+\boldsymbol{u}_0-\boldsymbol{u}_*)$, where $\boldsymbol{W}_2$ is another positive definite weighting matrix. Choose $\mathcal{J}_3 = \mathcal{J}_1 + \sigma\mathcal{J}_2$, where $0<\sigma\ll 1$ for prioritising $\mathcal{J}_1$. Further derive $\mathcal{J}_3$ as

$$\begin{aligned} \mathcal{J}_3 = \mathcal{J}_1+\sigma\mathcal{J}_2 =& \frac{1}{2}\Delta\boldsymbol{u}^{\mathrm{T}}\left(\bar{\boldsymbol{\mathcal{B}}}^{\mathrm{T}}(\boldsymbol{x}_0)\boldsymbol{W}_1\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)+\sigma\boldsymbol{W}_2\right)\Delta\boldsymbol{u} \\ &+\left((\boldsymbol{y}_0^{(\boldsymbol{\rho})}-\boldsymbol{\nu}_c)^{\mathrm{T}}\boldsymbol{W}_1\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)+(\boldsymbol{u}_0-\boldsymbol{u}_*)^{\mathrm{T}}\sigma\boldsymbol{W}_2\right)\Delta\boldsymbol{u} \\ &+\frac{1}{2}\left((\boldsymbol{y}_0^{(\boldsymbol{\rho})}-\boldsymbol{\nu}_c)^{\mathrm{T}}\boldsymbol{W}_1(\boldsymbol{y}_0^{(\boldsymbol{\rho})}-\boldsymbol{\nu}_c)+(\boldsymbol{u}_0-\boldsymbol{u}_*)^{\mathrm{T}}\sigma\boldsymbol{W}_2(\boldsymbol{u}_0-\boldsymbol{u}_*)\right) \end{aligned} \tag{III.1.9}$$

Since within every time step, $\boldsymbol{u}_0$ and $\boldsymbol{y}_0^{(\boldsymbol{\rho})}$ are measured, while $\boldsymbol{u}_*$ and $\boldsymbol{\nu}_c$ are constants, only the terms related to $\Delta\boldsymbol{u}$ need to be minimised. Therefore, the incremental control

allocation problem is formulated as:

$$\begin{aligned} \underset{\Delta \boldsymbol{u}}{\textbf{MINIMISE}} \quad & \mathcal{J}_4 = \frac{1}{2}\Delta \boldsymbol{u}^{\mathrm{T}} \left(\bar{\boldsymbol{\mathcal{B}}}^{\mathrm{T}}(\boldsymbol{x}_0) \boldsymbol{W}_1 \bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0) + \sigma \boldsymbol{W}_2 \right) \Delta \boldsymbol{u} \\ & + \left((\boldsymbol{y}_0^{(\boldsymbol{\rho})} - \boldsymbol{\nu}_c)^{\mathrm{T}} \boldsymbol{W}_1 \bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0) + (\boldsymbol{u}_0 - \boldsymbol{u}_*)^{\mathrm{T}} \sigma \boldsymbol{W}_2 \right) \Delta \boldsymbol{u} \\ \textbf{SUBJECT TO} \quad & \boldsymbol{A}_u \Delta \boldsymbol{u} \leq \boldsymbol{b}_u \end{aligned} \tag{III.1.10}$$

The optimisation problem formulated in Eq. (III.1.10) is convex because the objective function $\mathcal{J}_4$ is a convex function and the feasible set given by $\boldsymbol{A}_u \Delta \boldsymbol{u} \leq \boldsymbol{b}_u$ is a convex set [28]. The active-set solver is selected because of its superior performance on solving small to medium size quadratic programming problems [29]. In contrast to the $\Delta \boldsymbol{u}_{\text{indi-pi}}$ in Eq. (III.1.7), it is difficult to write an analytical expression for the control input given by quadratic programming. Consequently, Theorem 3 is not applicable here. To derive a sufficient condition for the boundedness of $\boldsymbol{\varepsilon}_{\text{indi}}$, when the quadratic programming allocator is applied, assume at every time step, $\boldsymbol{\mathcal{B}}(\boldsymbol{x}_0) = \boldsymbol{K}_{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)$, then the following theorem holds:

Theorem 4. *When the quadratic programming control allocation is used (Eq. (III.1.10)), if* $\|\boldsymbol{I} - \boldsymbol{K}_{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\|_2 \leq \bar{b}' < 1$, *and if* $\boldsymbol{\delta}(\boldsymbol{x}, \Delta t)$, $\Delta \boldsymbol{d}_y$, *and* $\boldsymbol{\varepsilon}_{ca}$ *are respectively bounded by* $\bar{\delta}$, $\overline{\Delta d}$, *and* $\bar{\varepsilon}_{ca}$, *then under sufficiently high sampling frequency,* $\boldsymbol{\varepsilon}_{indi}$ *in Eq. (III.1.5) is ultimately bounded.*

Proof: See Appendix.

III.1.2.3. VIRTUAL SHAPE FUNCTIONS

In the preceding subsections, the number of control inputs equals the number of servos. Although the relative command differences of any adjacent servos have been constrained by Eq. (III.1.8), the resulting $\boldsymbol{u} \in \mathbb{R}^{m \times 1}$ does not necessarily lead to a smooth wing shape. This subsection will introduce virtual shape functions to solve this problem.

Define a reference axis where O_s is located at the wing root, while $O_s x_s$ is aligned with the servo line (Fig. III.1.1). The aim is to make the morphing wing trailing-edge shape as close as possible to a smooth function $f_s(t, x_s) : [0, \infty) \times \mathbb{R} \to \mathbb{R}$. Referring to the Weierstrass theorem [30], when q is sufficiently large, any sufficiently smooth function can be approximated by a q-th order polynomial, i.e., $f_s(x_s, t) \approx \tilde{f}_s(x_s, t) = \boldsymbol{\Theta}^{\mathrm{T}}(t)\boldsymbol{\Phi}(x_s)$, with $\boldsymbol{\Theta}(t) : [0, \infty) \to \mathbb{R}^{q \times 1}$, $\boldsymbol{\Phi}(x_s) : \mathbb{R} \to \mathbb{R}^{q \times 1}$.

The Chebyshev polynomials are selected in this Chapter because of their nearly optimal property and orthogonality [31]. Hence, the virtual shape function is designed as $\boldsymbol{\Phi}(x_s) = [T_0(x_s), T_1(x_s), ..., T_q(x_s)]^{\mathrm{T}}$, whose elements are the Chebyshev polynomials of the first kind: $T_1(x_s) = 1$, $T_2(x_s) = x_s$, $T_{i+1} = 2x_s T_i(x_s) - T_{i-1}(x_s)$, $i = 2, 3, ..., q-1$. Consequently, any $\boldsymbol{\Theta}(t) = [\theta_1(t), \theta_2(t), ..., \theta_q(t)]^{\mathrm{T}}$ guarantees the q-th order smoothness of $\tilde{f}_s(t, x_s)$. Denote the servo spanwise location vector as $\boldsymbol{x}_s = [x_{s,1}, x_{s,2}, ..., x_{s,m}]^{\mathrm{T}}$, which can be normalised by the half-wing span L, yielding $\bar{\boldsymbol{x}}_s = [x_{s,1}/L, x_{s,2}/L, ..., x_{s,m}/L]^{\mathrm{T}}$.

III.1

Substituting the normalised servo location vector into $\tilde{f}_s(x_s,t)$ yields:

$$\begin{pmatrix} \tilde{f}_s(\bar{x}_{s,1},t) \\ \tilde{f}_s(\bar{x}_{s,2},t) \\ \vdots \\ \tilde{f}_s(\bar{x}_{s,m},t) \end{pmatrix} = \begin{bmatrix} T_0(\bar{x}_{s,1}) & T_1(\bar{x}_{s,1}) & \dots & T_q(\bar{x}_{s,1}) \\ T_0(\bar{x}_{s,2}) & T_1(\bar{x}_{s,2}) & \dots & T_q(\bar{x}_{s,2}) \\ \vdots & \vdots & & \vdots \\ T_0(\bar{x}_{s,m}) & T_1(\bar{x}_{s,m}) & \dots & T_q(\bar{x}_{s,m}) \end{bmatrix} \begin{pmatrix} \theta_1(t) \\ \theta_2(t) \\ \vdots \\ \theta_q(t) \end{pmatrix} \triangleq \boldsymbol{\Phi}_{\bar{x}_s} \boldsymbol{\Theta}(t) \tag{III.1.11}$$

where $\boldsymbol{\Phi}_{\bar{x}_s} \in \mathbb{R}^{m\times q}$ becomes a constant shape matrix. Essentially, $\boldsymbol{\Phi}_{\bar{x}_s}$ provides a mapping between a smooth wing shape and $\boldsymbol{\Theta}(t)$. In view of this, choose a new control vector $\boldsymbol{u}_v = \boldsymbol{\Theta}(t) \in \mathbb{R}^{q\times 1}$. If the actual control command is mapped as $\boldsymbol{u} = \boldsymbol{\Phi}_{\bar{x}_s} \boldsymbol{u}_v$, then this $\boldsymbol{u}$ can result in smooth wing shapes at all $t \in [0,\infty)$. The first five normalised virtual shape functions for the SmartX-Alpha are illustrated in Fig. III.1.2.

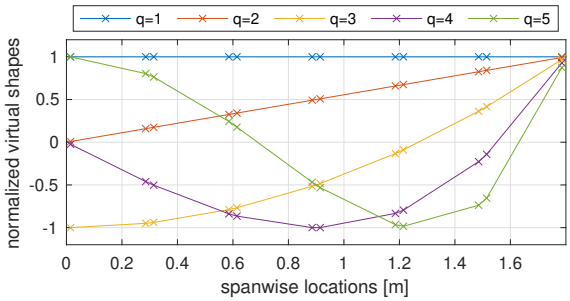


Figure III.1.2: Normalised virtual shape functions with markers indicating the SmartX-Alpha servo locations.

Because $\boldsymbol{\Phi}_{\bar{x}_s}$ is a constant matrix, this mapping also holds for the control increments, i.e., $\Delta\boldsymbol{u} = \boldsymbol{\Phi}_{\bar{x}_s}\Delta\boldsymbol{u}_v$. Actually, the control effective matrix with respect to $\Delta\boldsymbol{u}_v$ becomes $\bar{\boldsymbol{\mathcal{B}}}'(\boldsymbol{x}_0) = \left(\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\boldsymbol{\Phi}_{\bar{x}_s}\right) \in \mathbb{R}^{p\times q}$. If the column rank of $\bar{\boldsymbol{\mathcal{B}}}'$ is larger than p, then the quadratic programming problem integrated with virtual shape functions is formulated as

$$\begin{aligned} \underset{\Delta\boldsymbol{u}_v}{\textbf{MINIMISE}} \quad & \mathcal{J}_5 = \frac{1}{2}\Delta\boldsymbol{u}_v^{\mathrm{T}}\left(\boldsymbol{\Phi}_{\bar{x}_s}^{\mathrm{T}}\bar{\boldsymbol{\mathcal{B}}}^{\mathrm{T}}(\boldsymbol{x}_0)\boldsymbol{W}_1\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\boldsymbol{\Phi}_{\bar{x}_s} + \sigma\boldsymbol{\Phi}_{\bar{x}_s}^{\mathrm{T}}\boldsymbol{W}_2\boldsymbol{\Phi}_{\bar{x}_s}\right)\Delta\boldsymbol{u}_v \\ & \quad + \left((\boldsymbol{y}_0^{(\boldsymbol{\rho})} - \boldsymbol{\nu}_c)^{\mathrm{T}}\boldsymbol{W}_1\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0) + (\boldsymbol{u}_0 - \boldsymbol{u}_*)^{\mathrm{T}}\sigma\boldsymbol{W}_2\right)\boldsymbol{\Phi}_{\bar{x}_s}\Delta\boldsymbol{u}_v \\ \textbf{SUBJECT TO} \quad & \left(\boldsymbol{A}_u\boldsymbol{\Phi}_{\bar{x}_s}\right)\Delta\boldsymbol{u}_v \le \boldsymbol{b}_u \end{aligned} \tag{III.1.12}$$

Essentially, the optimisation problem in Eq. (III.1.12) is transformed from Eq. (III.1.10) using the mapping $\Delta\boldsymbol{u} = \boldsymbol{\Phi}_{\bar{x}_s}\Delta\boldsymbol{u}_v$. It can be verified that Eq. (III.1.12) formulates a convex optimisation problem because $\mathcal{J}_5$ is a convex function, and because the feasible set $\left(\boldsymbol{A}_u\boldsymbol{\Phi}_{\bar{x}_s}\right)\Delta\boldsymbol{u}_v \le \boldsymbol{b}_u$ is a convex set [28]. Furthermore, because the dimension of $\Delta\boldsymbol{u}_v$ is lower than that of $\Delta\boldsymbol{u}$, the computational load is also reduced by introducing the virtual shape functions. The corollary of Theorem 4 is given:

Corollary 1. *When the quadratic programming control allocation with virtual shape functions is used (Eq. (III.1.12)), if* $\|\boldsymbol{I} - \boldsymbol{K}_{\mathcal{B}}(\boldsymbol{x}_0)\|_2 \le \bar{b}' < 1$, *and if* $\boldsymbol{\delta}(\boldsymbol{x},\Delta t)$, $\Delta\boldsymbol{d}_y$, *and* $\boldsymbol{\varepsilon}_{ca}$ *are respectively bounded by* $\bar{\delta}$, $\overline{\Delta d}$, *and* $\bar{\varepsilon}_{ca}$, *then under sufficiently high sampling frequency,* $\boldsymbol{\varepsilon}_{indi}$ *in Eq. (III.1.5) is ultimately bounded.*

Proof: See Appendix III.A1.2.

Remark 3. The virtual shape functions were also used in [16, 32] intending to address the relative deflection constraints. However, the usage of virtual shape itself is not sufficient for meeting the relative position constraints. By contrast, the control allocator formulated in Eq. (III.1.12) not only explicitly considers the position, rate, and relative position constraints but also leads to a smooth wing shape at every moment.

III.1.3. Experimental Results

In this section, the proposed incremental control will be applied to the SmartX-Alpha load alleviation problems. The experiment setup will be presented in Sec. III.1.3.1, following which the challenges in the experiment will be presented in Sec. III.1.3.2. The experimental results for manoeuvre load alleviation, gust load alleviation, as well as simultaneous gust and manoeuvre load alleviation, will be shown in Sec. III.1.3.3-III.1.3.5.

III.1.3.1. Experiment Setup

The experiments were conducted in the OJF wind tunnel of the Delft University of Technology. A two-vane gust generator is installed to produce aerodynamic disturbances at various magnitudes and frequencies. Additionally, to support the identification and structural cauterisation of the morphing wing, Ground Vibration Test (GVT) was performed prior to the wind tunnel experiments. The results of the modal testing campaign are presented in Appendix III.A1.1.

The SmartX-Alpha wing has twelve independent servos (Fig. III.1.1), thus $m = 12$. In order to alleviate the excessive loads (no matter caused by gusts or manoeuvres) without degrading the rigid-body command tracking performance, the load alleviation problems are converted to load reference tracking problems. As discussed in Sec. III.1.2, the $\boldsymbol{y}$ in Eq. (III.1.1) can be a function of any subset of the physical measurable outputs. For load alleviation purposes, choose $\boldsymbol{y} = [\int F_y, \int M_x]^{\mathrm{T}}$, where F_y and M_x are the measured wing root shear force and bending moment, respectively (Fig. III.1.3). Referring to the Theodorsen's theory [33], given a control surface deflection (a camber morphing for SmartX-Alpha), half of the circulatory lift gradually builds up, while the rest happens instantaneously. Therefore, a change in wing camber has direct influence on loads. Accordingly, for the selected inputs and outputs, the vector relative degree is $\boldsymbol{\rho} = [1, 1]^{\mathrm{T}}$.

Recall Sec. III.1.2, the only model information needed by INDI is the estimated control effectiveness matrix $\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)$. For the selected input and output vectors, $\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0) \in \mathbb{R}^{2\times 12}$. In theory, $\bar{\boldsymbol{\mathcal{B}}}$ is a function of states. Nonetheless, as has been proved by Theorems 3 and 4, the INDI control can passively resist a wide range of model uncertainties in $\bar{\boldsymbol{\mathcal{B}}}$. Therefore, in the experiment, a constant $\bar{\boldsymbol{\mathcal{B}}}$ matrix identified in the trimmed condition was consistently used by the controller. In this way, the control implementation process was simplified; the robustness of the controller was also tested.

In Eq. (III.1.4), the gain matrix is chosen as $\boldsymbol{K} = \mathrm{diag}\{0.1, 0.1\}$. The position constraints for the servos are $\boldsymbol{u}_{\max} = \boldsymbol{I}_{12\times 1} \cdot 30$ deg, $\boldsymbol{u}_{\min} = -\boldsymbol{I}_{12\times 1} \cdot 30$ deg. The rate constraints for the servos are $\underline{\boldsymbol{u}}_{\text{rate}} = -\boldsymbol{I}_{12\times 1} \cdot 80$ deg/s, $\bar{\boldsymbol{u}}_{\text{rate}} = \boldsymbol{I}_{12\times 1} \cdot 80$ deg/s. The relative position constrain vector $\bar{\boldsymbol{u}}_{\text{adj}} \in \mathbb{R}^{11\times 1}$. For $i = 1, 2, ..., 11$, when i is an odd

III.1

number, $\bar{u}_{\text{adj},i} = 55$ deg; otherwise, $\bar{u}_{\text{adj},i} = 10$ deg. In Eq. (III.1.10), σ is chosen as 0.001 to prioritise $\mathcal{J}_1$. The weighting matrices are chosen as $\boldsymbol{W}_1 = \boldsymbol{I}_{2\times 2}$ and $\boldsymbol{W}_2 = \boldsymbol{I}_{12\times 12}$. A block diagram for the experiment setup is presented in Fig. III.1.3.

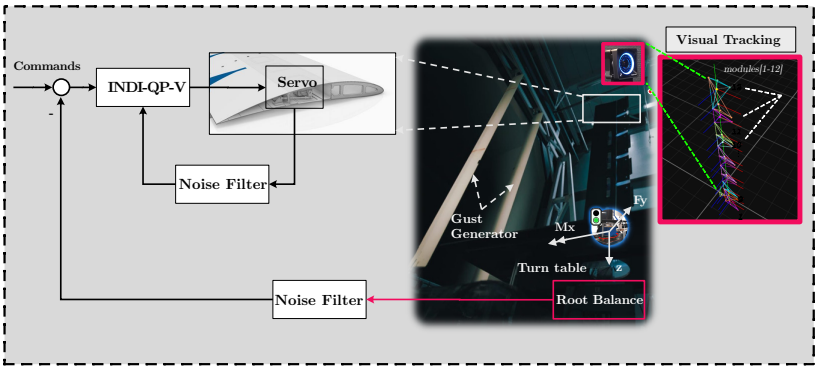


Figure III.1.3: A block diagram for experiment setup.

As shown in Fig. III.1.3, the SmartX-Alpha wing was vertically mounted on a turn table. The operational point was selected as $V = 15$ m/s, $\alpha = -2.89$ deg (turn table angle equals 1.00 deg). The three-axes root reaction forces and moments were measured by the OJF External Balance in 1000 Hz. The core component of this balance is a set of strain gauges. For real-world aircraft, strain gauges can also be attached to wing-root structures to provide root reaction forces for feedback control. All twelve servos were connected to an RS-485 device, communicating serially over the physical USB bus updating at 66.7 Hz. The communication delay was approximately 15 ms. The wing displacements were captured by a visual tracking system (OptiTrack) [34]. The local wing loads were measured by embedded strain gauges.

III.1.3.2. Practical Issues

Nonlinear Backlash

Backlash is a clearance or lost motion phenomenon in mechanical systems caused by gaps between the mechanical components. Consider a general mechanical linkage; denote the generalised displacement of the driving and driven part as u and τ, respectively. The widely adopted free-play model entails: if $u < u_{f_-}, \tau = k_1(u - u_{f_-})$; if $u > u_{f_+}, \tau = k_2(u - u_{f_+})$; otherwise, $\tau = 0$ [35]. $k_1 > 0, k_2 > 0$ are the linear slopes; $u_{f_+} > 0$ and $u_{f_-} < 0$ represent the free-play deadband. Actuator free-play can lead to limit cycle oscillations [36]. The backlash nonlinearity is even more challenging [37]:

$$\dot{\tau} = f(\tau, u, \dot{u}) = \begin{cases} k_1\dot{u}, & \text{if } \dot{u} < 0 \text{ and } \tau = k_1(u - u_{f_-}) \\ k_2\dot{u}, & \text{if } \dot{u} > 0 \text{ and } \tau = k_2(u - u_{f_+}) \\ 0, & \text{otherwise} \end{cases} \tag{III.1.13}$$

Equation (III.1.13) presents a velocity-driven dynamic system. Different from the free-play, τ in Eq. (III.1.13) is also dependent on the history of u. This hysteresis effect was

also observed during the experiment. In Fig. III.1.4, all the twelve servos execute the same command: starts from 30 deg and gradually reduces to -30 deg (surfaces morph upwards), and then gradually increases back to 30 deg (surfaces morph downwards). Figure III.1.4 shows that due to backlash, the same servo angle settings lead to different force responses in upstroke and downstroke.

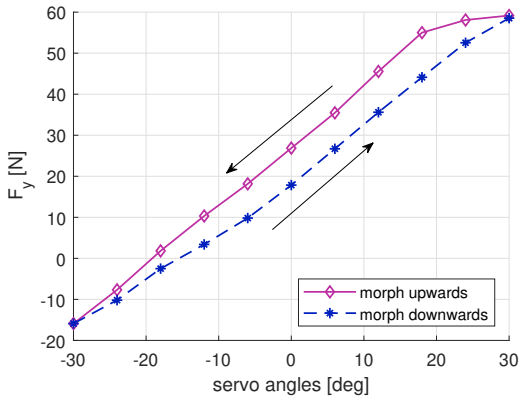


Figure III.1.4: Backlash-induced wing root shear force hysteresis loop in the experiment.

The SmartX-Alpha is the first prototype featuring the distributed morphing TRIC concept. The manufacturing and integration process involved largely handcrafted structural components and manual laminate layup, which inevitably led to manufacturing imperfections. One such imperfection was the exact tolerance between the skin and the sliding interface. This gap was filled with a spacer that added additional friction. Combined with slack in the actuator mechanism and a relatively large stiffness gap between the rigid aluminium pick-up point and its attachment to the flexible skin, the pick-up point exhibited local out-of-plane rotations. Moreover, the bottom skin exhibited local bending motions. Consequently, whenever the servo command changes direction, the pick-up point needs to rotate, and the bottom skin needs to bend a little before the ideal translational sliding actually happens. These phenomena were only discovered during the tightly-scheduled experiment and were not foreseen by the control designs. Therefore, it was decided to test the robustness of the controller to backlash and friction in this experiment.

Actuator Dynamics and Fault

The servos of the SmartX-Alpha are the Volz DA 22-12-4112 [7]. To identify the servo dynamics, a sweep signal with a magnitude of ± 30 deg was given to the servo. By analyzing the input and output signals, is was identified that the second-order system $H(s) = \frac{\omega^2}{s^2+2\zeta\omega s+\omega^2}$ can represent the servo dynamics. The identified parameters are $\zeta = 0.71$, $\omega = 16.52$ rad/s. Consequently, the cut-off frequency of the servo equals 16.35 rad/s (2.60 Hz).

After conducting the control effectiveness identification and before implementing the controllers, the 9$^{\text{th}}$ actuator was non-operational. This failure resulted from adhesive bond failure between the aluminium pick-up point and the composite morphing skin. Conse-

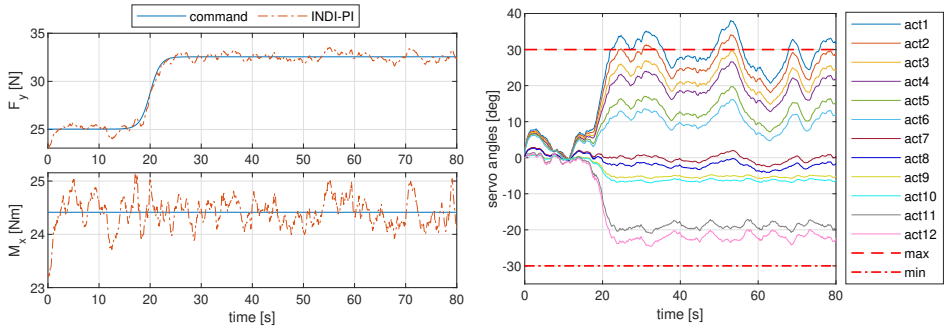
quently, the control effectiveness of the 9th actuator becomes zero. Moreover, since shear forces can still propagate within module four via the composite shell, and propagate to the adjacent module via the elastomer, the control effectiveness of the 8th and 10th actuators were also affected. The repair would require unmounting the wind tunnel setup, extracting the morphing trailing edge from the wing structure, and waiting for a new adhesive layer to cure. Given the time constraints, a choice was made to disable the 9th servo and test the robustness of the controller to actuator failures. Also, the control effectiveness identified in the healthy condition was still used in the implementation.

Coloured Noise

The signals provided by the root balance contain measurement noise. Experimental results show that the measurement noises of F_y and M_x are coloured and also contain considerable energy in the low-frequency range. To reduce the noise energy, the second-order low-pass filter with transfer function $H(s) = \frac{\omega^2}{s^2+2\zeta\omega s+\omega^2}$ is selected. Choosing the filter parameters is a trade-off: a low cut-off frequency leads to better noise attenuation but causes a larger phase lag in the closed-loop system. After several experimental trials, the parameters of the noise filter were chosen as $\zeta = 0.8$, $\omega = 10$ rad/s. Consequently, the noise filter cut-off frequency equals 8.67 rad/s (1.38 Hz).

III.1.3.3. Manoeuvre Load Alleviation

In this subsection, the manoeuvre load alleviation performance of the proposed controller will be evaluated experimentally. An aircraft symmetric pull-up manoeuvre is considered. The control objective is to increase lift while reducing wing root bending moment by spanwise lift redistribution. In Fig. III.1.5, F_y is commanded to increase by 30 %, while M_x is

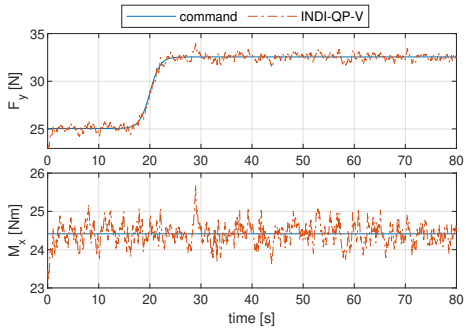


(a) Load responses.

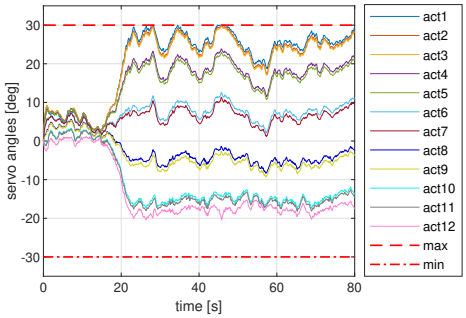
(b) Servo angle commands.

Figure III.1.5: Manoeuvre load alleviation performance of INDI with pseudo inverse control allocation.

commanded to remain at its trimmed value. A sigmoid function is adopted for a smooth command transition. Figure III.1.5a shows that the load commands are tracked in spite of actuator fault, delay, and backlash. Moreover, as illustrated in Fig. III.1.5b, the servo at the wing tip (12th) receives negative command (making the wing morph upwards), while the servo command gradually increases from the wing tip to the root. As a consequence, the wing aerodynamic centre is moved inboard by the trailing-edge morphing.



(a) Load responses.

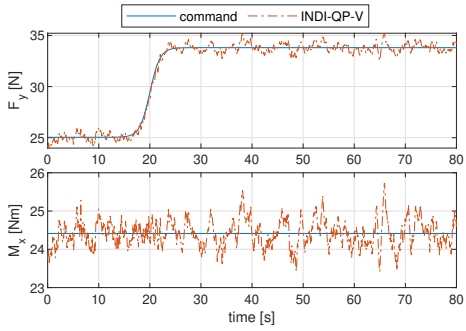


(b) Servo angle commands.

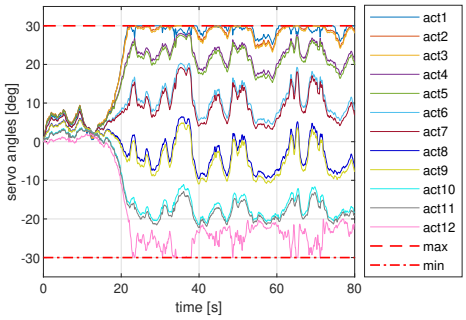
Figure III.1.6: Manoeuvre load alleviation performance of INDI using quadratic programming and virtual shapes.

However, neither input constraint nor spanwise servo location is considered in this pseudo inverse control allocation (Sec. III.1.2.2). Two drawbacks are identified: first, the hardware constraints can be violated (1st and 2nd servos in Fig. III.1.5b); second, it can cause high tension in the elastomer. For example, at $t = 53.3$ s, the command difference between the 6th and 7th servos are 14.2 deg. However, the spanwise distance between these two servos is only 29.0 mm. This rapid angle change in a short distance can overstretch the elastomer. These two drawbacks are overcome in INDI-QP-V, which explicitly considers input constraints and ensures the wing smoothness. Figure III.1.6a shows that INDI-QP-V increases F_y by 30 % without amplifying M_x. Figure III.1.6b confirms that the input constraints are not violated and the inter-modular command gaps are much smaller than the case in Fig. III.1.5b.

To further demonstrate the effectiveness of INDI-QP-V, the controller is asked to increase F_y by 35 % without raising M_x. Since this load alleviation task is more challenging than the previous one, the servo angle commands in Fig. III.1.7b are also saturated more frequently. Nevertheless, the input constraints are not violated; the inter-modular transitions are smooth; the load alleviation mission is also achieved (Fig. III.1.7a).



(a) Load responses.



(b) Servo angle commands.

Figure III.1.7: Performance of INDI-QP-V in a challenging manoeuvre load alleviation task.

Figure III.1.8 shows that at the majority of time span, $\|\boldsymbol{\varepsilon}_{\text{ca}}\|_2 \leq 1 \times 10^{-3}$. When severe saturation occurs, $\|\boldsymbol{\varepsilon}_{\text{ca}}\|_2$ is still bounded by 0.12. Moreover, the control allocator converges within one iteration when there is no saturation, and converges within ten iterations when saturation occurs. In all cases, the computational load is low, and the control commands are realised in real-time.

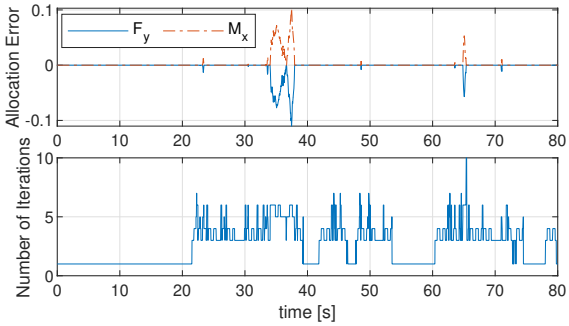


Figure III.1.8: Allocation errors and number of iterations of INDI-QP-V in a challenging MLA task.

III.1.3.4. Gust Load Alleviation

In Sec. III.1.3.3, experimental results have demonstrated that INDI-QP-V performance is superior to INDI-PI. This subsection will present the gust load alleviation effectiveness of INDI-QP-V. To generate the "1-cos" gust, the rotational angle of each gust generator vane obeys: $\theta(t) = A_g(1-\cos(2\pi f_g t+\phi))$, where ϕ is the phase shift. The corresponding gust angle is $\alpha_g(t) = (A_g/2)(1-\cos(2\pi f_g(t-d_{gw}/V)+\phi))$, where d_{gw} represents the gust travel distance; V is the nominal wind speed. To test robustness, the gust information was kept unknown to the controller.

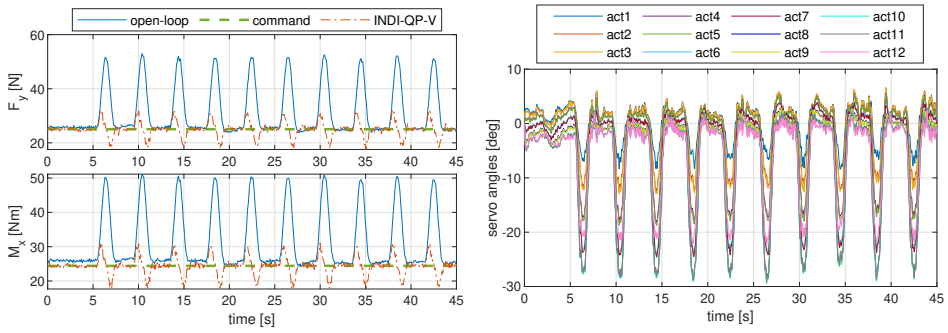
Denote the references for F_y and M_x as F_{y_*} and M_{x_*}, respectively. Four performance metrics are introduced: 1) the reduction rate of the maximum value of $F_y - F_{y_*}$; 2) the reduction rate of the root-mean-square (RMS) value of $F_y - F_{y_*}$; 3) the reduction rate of the maximum value of $M_x - M_{x_*}$; 4) the reduction rate of the RMS value of $M_x - M_{x_*}$. Take the last performance metric as an example, the reduce rate is calculated as $\frac{(\text{RMS}(M_x-M_{x_*}))|_{\text{open}}-(\text{RMS}(M_x-M_{x_*}))|_{\text{closed}}}{(\text{RMS}(M_x-M_{x_*}))|_{\text{open}}}$, where $(\cdot)|_{\text{open}}$ and $(\cdot)|_{\text{closed}}$ respectively means evaluating $(\cdot)$ in the open-loop or closed-loop condition.

In Fig. III.1.9, the gust generator motions obey $A_g = 3.5$ deg and $f_g = 0.5$ Hz. In the open-loop case, the maximum load increments in F_y and M_x are 27.85 N and 26.72 N·m, respectively. By using INDI-QP-V, these values are reduced to 6.88 N and 6.64 N·m. Over 75 % of reductions are achieved in all four performance metrics (Table III.1.1). Figure III.1.9b shows that the inter-modular transitions are smooth and no saturation occurs. Because of the coloured measurement noises, the measured load variations are non-zero even without gust. When these relatively small variations are fed back to the controller, small oscillatory commands are generated. However, due to backlash (Sec. III.1.3.2), a servo angle change within the deadband has no effect on the morphing surface, which further results in null load change. At the next time step, when the controller "sees" that

Table III.1.1: Gust load reduction rate using INDI-QP-V at various frequencies.

Frequency [Hz]	max$(F_y - F_{y_*})$	RMS$(F_y - F_{y_*})$	max$(M_x - M_{x_*})$	RMS$(M_x - M_{x_*})$
0.5	75.57 %	76.41 %	75.16 %	77.39 %
1.0	56.25 %	53.73 %	57.04 %	56.13 %
1.5	47.86 %	40.80 %	47.87 %	43.28 %
2.0	40.52 %	29.23 %	40.47 %	32.58 %
2.5	25.25 %	19.49 %	27.34 %	24.23 %
3.0	15.26 %	14.83 %	18.53 %	20.77 %
3.5	7.52 %	6.12 %	7.14 %	14.29 %
4.0	8.83 %	-1.44 %	10.28 %	6.98 %
4.5	-0.96 %	-6.77 %	5.79 %	4.24 %

the previous command has no effect, a command with a higher magnitude will be given to the servo until it moves out of the deadband. These are the physical explanations for the high-frequency oscillations in Fig. III.1.9b.



(a) Load responses.

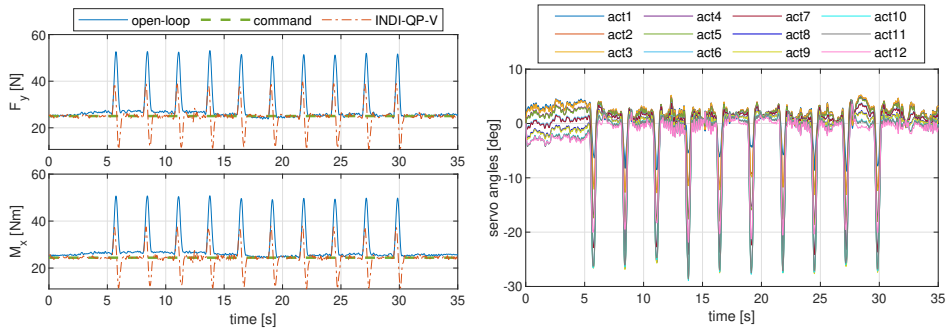
(b) Servo angles.

Figure III.1.9: Load alleviation performance of INDI-QP-V under 0.5 Hz gusts.

Figure III.1.10 illustrates the open- and closed-loop load responses when $A_g = 3.5$ deg and $f_g = 1.5$ Hz. In the open-loop case, the maximum load increments in F_y and M_x are 28.09 N and 26.36 N·m, respectively. With the help of INDI-QP-V, these values are respectively reduced to 14.65 N and 13.74 N·m. Table III.1.1 shows that more than 40 % of load reductions are achieved in all four performance metrics. Comparing Fig. III.1.10 with Fig. III.1.9, we can see that the alleviation performance degrades with the increase in gust frequency. Moreover, the closed-loop load responses in Fig. III.1.10a are more lagged behind than those in Fig. III.1.9a.

Table III.1.1 summarises the load reduction rates of INDI-QP-V. Note in all cases, the control gains and A_g remain consistent. The reduction rates are over 75 % percent when $f_g = 0.5$ Hz, but reduces to around 20 % when $f_g = 2.5$ Hz. When f_g further increases to 4.5 Hz, the RMS value of $F_y - F_{y_*}$ is even higher in the closed-loop condition. The main reason for this performance degradation is the phase lag in the closed-loop system. Recall Sec. III.1.3.2, the cut-off frequencies of the servo and the noise filter are 2.60 Hz and

III.1



(a) Load responses.

(b) Servo angles.

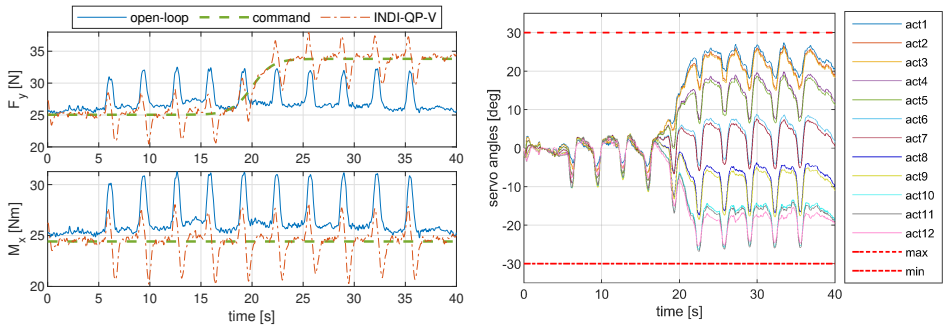
Figure III.1.10: Load alleviation performance of INDI-QP-V under 1.5 Hz gusts.

1.38 Hz, respectively. These lead to large phase lags in the high-frequency range, which further results in the performance deterioration. To improve the performance, we can use less noisy sensors and faster servos, and disclose the gust information to the controller if onboard gust sensing is available.

III.1.3.5. Simultaneous Gust and Manoeuvre Load Alleviation

In the literature, GLA and MLA are usually seen as two research topics. However, during real flights, instead of classifying loads by their causes, it is more meaningful to achieve the necessary loads for performing manoeuvres while neutralising the excessive loads (no matter induced by manoeuvres or gusts). INDI-QP-V is a good candidate to achieve this goal. As presented in Sec. III.1.3.1, the real-world load alleviation task is seen as a load command tracking problem by INDI-QP-V. Consequently, by minimising the error between the commanded and real loads, simultaneous gust and manoeuvre load alleviation can be realised. This design also ensures the task applicability. In fact, in the experiments of MLA (Sec. III.1.3.3), GLA (Sec. III.1.3.4), and simultaneous GLA and MLA (Sec. III.1.3.5), only the load commands are task-dependent; there is no need to change the control architecture nor the control parameters.

Figure III.1.11 presents the experimental results for simultaneous gust and manoeuvre load alleviation. F_y is commanded to increase by 35 % for achieving a pull-up manoeuvre, while M_x is asked to stay at its nominal value in spite of the manoeuvre and gusts ($A_g = 1$ deg and $f_g = 1$ Hz). Figure III.1.11a demonstrates that INDI-QP-V is able to alleviate the excessive loads. Using the performance metrics (Sec. III.1.3.4), the maximum and RMS values of $F_y - F_{y_*}$ are respectively reduced by 44.31 % and 67.76 %; the maximum and RMS values of $M_x - M_{x_*}$ are reduced by 45.58 % and 46.35 %, respectively. After $t = 17$ s, the outboard wing starts to morph upwards while the inboard wing begins to morph downwards for spanwise lift redistribution (Fig. III.1.11b). Moreover, on top of the redistributive motions, the wing actively morphs upwards to reduce the gust-induced loads. Furthermore, the quadratic programming control allocator ensures no saturation occurs; virtual shape functions realise smooth wing shape at every moment.



(a) Load responses.

(b) Servo angles.

Figure III.1.11: Simultaneous manoeuvre and gust load alleviation performance of INDI-QP-V.

III.1.4. Comparisons with Linear Quadratic Gaussian Control

In Sec. III.1.3, experimental results have demonstrated the effectiveness of INDI-QP-V in GLA, MLA and simultaneous GLA and MLA tasks. According to the literature, the linear quadratic Gaussian (LQG) control is one of the most popular methods for load alleviation [11, 15, 32, 38]. Therefore, the proposed INDI-QP-V control will be compared to LQG control.

III.1.4.1. LQG Control Design

The LQG control is essentially a combination of a Kalman filter for state estimation and a linear-quadratic regulator (LQR) for stabilisation. The linearised SmartX-Alpha dynamics are: $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{B}_g\alpha_g$, $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}$, where α_g is the gust input angle. $\boldsymbol{y} = [\boldsymbol{y}_b^{\mathrm{T}}, \boldsymbol{y}_a^{\mathrm{T}}]^{\mathrm{T}}$, with $\boldsymbol{y}_b$ includes F_y and M_x, and $\boldsymbol{y}_a$ denotes the wing acceleration measurements.

First, assume the states are known, design an LQR to make $\boldsymbol{y}_b$ track its reference $\boldsymbol{y}_r$. The LQR design requires the estimated system model: $\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}, \bar{\boldsymbol{C}}, \bar{\boldsymbol{D}}$. In view of the benefits of using virtual shape functions (Sec. III.1.2.3), the following transformation is also adopted by LQR: $\boldsymbol{u} = \boldsymbol{\Phi}_{\bar{x}_s}\boldsymbol{u}_v$. Design an LQR for the following augmented system:

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \boldsymbol{y}_b - \boldsymbol{y}_r \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{A}} & \boldsymbol{0} \\ \bar{\boldsymbol{C}}_b & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \int(\boldsymbol{y}_b - \boldsymbol{y}_r) \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{B}} \\ \bar{\boldsymbol{D}}_b \end{bmatrix} \boldsymbol{\Phi}_{\bar{x}_s}\boldsymbol{u}_v + \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{y}_r \end{bmatrix} \tag{III.1.14}$$

where $\bar{\boldsymbol{C}}_b$ and $\bar{\boldsymbol{D}}_b$ respectively equal the first two rows of $\bar{\boldsymbol{C}}$ and $\bar{\boldsymbol{D}}$ (corresponding to $\boldsymbol{y}_b$). Define the augmented state vector as $\boldsymbol{X} = [\boldsymbol{x}^{\mathrm{T}}, \int(\boldsymbol{y}_b - \boldsymbol{y}_r)^{\mathrm{T}}]^{\mathrm{T}}$. Note that Eq. (III.1.14) can be written as $\dot{\boldsymbol{X}} = \boldsymbol{A}_{\mathrm{aug}}\boldsymbol{X} + \boldsymbol{B}_{\mathrm{aug}}\boldsymbol{u}_v + \boldsymbol{y}_{r,\mathrm{aug}}$. Minimising the cost function, given by $\mathcal{J}_6 = \lim \frac{1}{2}\int_0^\infty [\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{X} + \boldsymbol{u}_v^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}_v]\,\mathrm{d}t$, yields optimal control input $\boldsymbol{u}_v = \boldsymbol{K}_X\boldsymbol{X} + \boldsymbol{K}_r\boldsymbol{y}_{r,\mathrm{aug}}$, $\boldsymbol{K}_X = -\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{aug}}^{\mathrm{T}}\boldsymbol{S}$, $\boldsymbol{K}_r = -\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{aug}}^{\mathrm{T}}(\boldsymbol{S}\boldsymbol{B}_{\mathrm{aug}}\boldsymbol{R}^{-1}\boldsymbol{B}_{\mathrm{aug}}^{\mathrm{T}} - \boldsymbol{A}_{\mathrm{aug}}^{\mathrm{T}})^{-1}\boldsymbol{S}$, in which $\boldsymbol{S}$ is the solution of the associated Riccati equation.

Second, design a Kalman filter for $\dot{\boldsymbol{x}} = \bar{\boldsymbol{A}}\boldsymbol{x} + \bar{\boldsymbol{B}}\boldsymbol{u} + \bar{\boldsymbol{G}}\boldsymbol{w},\ \boldsymbol{y} = \bar{\boldsymbol{C}}\boldsymbol{x} + \bar{\boldsymbol{D}}\boldsymbol{u} + \bar{\boldsymbol{H}}\boldsymbol{w} + \boldsymbol{v}$. The process noise $\boldsymbol{w}$ and measurement noise $\boldsymbol{v}$ are assumed to be white. They also satisfy $E(\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}) = \boldsymbol{Q}_k$, $E(\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}) = \boldsymbol{R}_k$, $E(\boldsymbol{w}\boldsymbol{v}^{\mathrm{T}}) = \boldsymbol{N}_k$. Subsequently, design a dynamic system $\dot{\hat{\boldsymbol{x}}} = \bar{\boldsymbol{A}}\hat{\boldsymbol{x}} + \bar{\boldsymbol{B}}\boldsymbol{u} + \boldsymbol{L}\left(\boldsymbol{y} - \bar{\boldsymbol{C}}\hat{\boldsymbol{x}} - \bar{\boldsymbol{D}}\boldsymbol{u}\right)$, where $\boldsymbol{L}$ is the optimal Kalman gain; this gives $\hat{\boldsymbol{x}} \to \boldsymbol{x}$ as $t \to \infty$.

Finally, integrate the LQR controller with the Kalman filter state observer, the resulting LQG control input is $\boldsymbol{u} = \boldsymbol{\Phi}_{\bar{x}_s}\boldsymbol{u}_v = \boldsymbol{\Phi}_{\bar{x}_s}\boldsymbol{K}_X[\hat{\boldsymbol{x}}^{\mathrm{T}}, \int(\bar{\boldsymbol{C}}_b\hat{\boldsymbol{x}} + \bar{\boldsymbol{D}}_b\boldsymbol{u} - \boldsymbol{y}_r)^{\mathrm{T}}]^{\mathrm{T}} + \boldsymbol{\Phi}_{\bar{x}_s}\boldsymbol{K}_r\boldsymbol{y}_{r,\mathrm{aug}}$.

III.1.4.2. Theoretical Comparisons

The LQG bears significant differences from the nonlinear INDI-QP-V control method. Although LQG can be applied to nonlinear systems, closed-loop stability is only guaranteed locally. Extending LQG to a broader state definition domain requires the gain-scheduling method. However, the gain-scheduled LQG is tedious to tune; its stability also heavily depends on the linearisation density and cannot be ensured in general cases. Another notable difference between the sensor-based incremental control is the strong robustness against uncertainties and disturbances demonstrated in various studies [22, 27, 39]. The robustness of LQG to model uncertainties and external disturbances can only be considered when implemented in a robust control framework such as H_∞ [12, 39, 40]. Additionally, methods such as the Loop Transfer Recovery (LTR) are suitable to enhance its robust stability [41]

In reality, the tedious system identification of the complete system dynamics (for $\bar{\boldsymbol{A}}$, $\bar{\boldsymbol{B}}$, $\bar{\boldsymbol{C}}$, $\bar{\boldsymbol{D}}$) and tuning processes were the main barriers in LQG implementation. When an LQR controller was designed based on the identified model and then integrated with a Kalman filter, the resulting LQG performed poorly in our experiment. The identification and tuning took much longer than planned, and the implemented LQG was not successful within the time limit.

The only model information needed by INDI-QP-V is the control effectiveness matrix $\bar{\boldsymbol{\mathcal{B}}}$. Due to this reason, the identification of the $\bar{\boldsymbol{\mathcal{B}}}$, gain tuning, and the entire hardware implementation of INDI-QP-V on the SmartX-Alpha was achieved within a day. Therefore, these evaluations have highlighted the benefits and ease of implementation of the sensor-based INDI-QP-V.

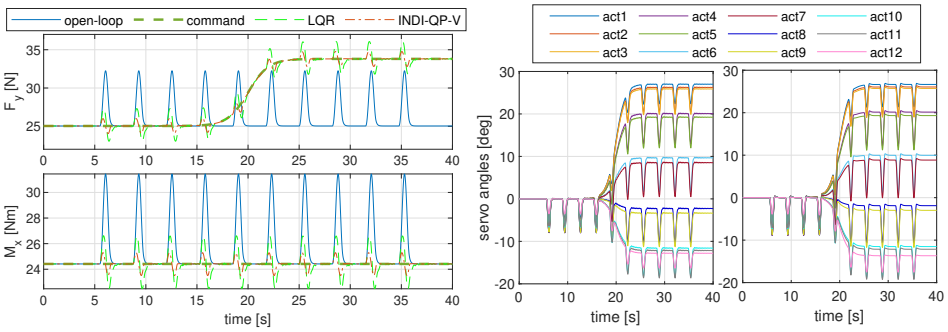
III.1.4.3. Load Alleviation Performance Comparisons

Since the LQG control did not work in the experiment within the time limit owing to its tedious model identification and tuning processes. There is no valid experimental data for LQG. In this subsection, the performance of LQG and INDI-QP-V is compared in the simulation environment. The simulation model was identified from the experimental data. The control parameters of INDI-QP-V are kept the same as those used in the experiment.

The comparisons start with an ideal case, where measurement noise, actuator fault and backlash are not included yet. More importantly, the state information is assumed to be known (LQG degrades to LQR). $\boldsymbol{Q}$ is designed as a partitioned matrix, with the upper left matrix equals $\boldsymbol{C}_b^{\mathrm{T}}\boldsymbol{C}_b$, the lower right matrix equals $10 \cdot \boldsymbol{I}_{2\times 2}$, and the rests are equal to zero. $\boldsymbol{R} = 260 \cdot \boldsymbol{I}_{5\times 5}$.

In Fig. III.1.12 the aircraft is asked to perform a pull-up manoeuvre in a gust field ($A_g = 1$ deg and $f_g = 1$ Hz). Although both controllers can make the wing follow the

load commands, INDI-QP-V has better load alleviation performance. The performance metrics are summarised in the second row of Table III.1.2 (Sim no noise (Fig. III.1.12)). It can be seen that the reduction rate of LQG is above 63 % while INDI-QP-V reduces loads by more than 85 %.



(a) Load responses.

(b) Servo angles: LQR (left) and INDI-QP-V (right).

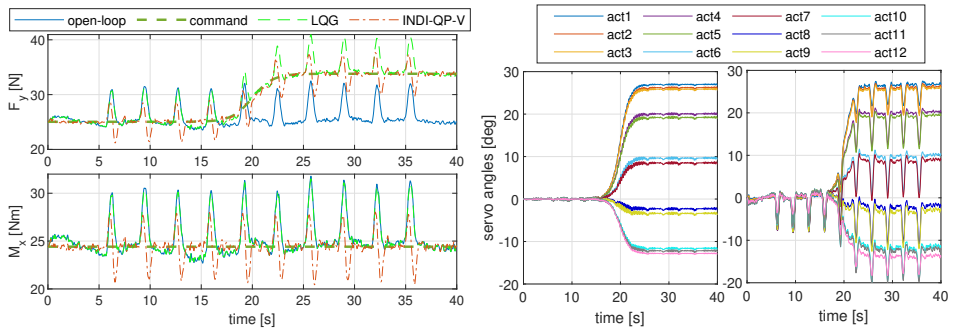
Figure III.1.12: Simultaneous gust and manoeuvre load alleviation without measurement noise.

Table III.1.2: Simultaneous gust and manoeuvre load reduction rate of INDI-QP-V and LQG.

Conditions	MAX($F_y - F_{y_*}$)		RMS($F_y - F_{y_*}$)		MAX($M_x - M_{x_*}$)		RMS($M_x - M_{x_*}$)	
	INDI-QP-V	LQG	INDI-QP-V	LQG	INDI-QP-V	LQG	INDI-QP-V	LQG
Exp. (Fig. III.1.11)	44.31 %	-	67.76 %	-	45.58 %	-	46.35 %	-
Sim n.noise (Fig. III.1.12)	86.65 %	68.25 %	93.95 %	85.44 %	87.48 %	68.60 %	85.71 %	63.73%
Sim noise (Fig. III.1.13)	39.99 %	-11.41 %	73.99 %	63.19 %	50.90 %	3.71 %	35.95 %	4.19 %
Sim noise, fault, + backlash (Fig. III.1.14)	25.99 %	-22.91 %	68.08 %	59.47 %	38.74 %	-12.56 %	19.21 %	-14.24 %

In the second comparison case, coloured measurement noises collected from the experiments are added. $\boldsymbol{R}_k$ is directly calculated using the applied noise values. Nevertheless, $\boldsymbol{Q}_k$ and $\boldsymbol{N}_k$ are difficult to tune because the uncertainties and gusts are far away from white noise. Their implemented values are $\boldsymbol{N}_k = 10^{-5}\cdot[3.16, 3.16, 6.41, 6.41, 87.1, 87.1]^{\mathrm{T}}$, $\boldsymbol{Q}_k = 1.02 \times 10^{-5}$. As shown in Table III.1.2 and Fig. III.1.13, mainly due to the phase lag induced by noise filtering, the minimum load reduction rate of INDI-QP-V reduces to 35.95 %. The performance of LQG is even worse: the maximum value of $F_y - F_{y_*}$ is even amplified by 11.41 % percent. As illustrated in Fig. III.1.13b, although phase lag exists, INDI-QP-V actively makes the wing morph upwards to reduce the gust loads. By contrast, although LQG can alleviate the manoeuvre load by spanwise lift redistribution, it is not effective in alleviating the gust loads. To obtain better gust load alleviation performance, LQG has to be used along with some additional disturbance estimators (e.g., disturbance observer [32]).

Apart from coloured noises, actuator fault and backlash are also added to the last comparison case. Owing to the pick-up point failure, the 9$^{\text{th}}$ actuator effectiveness equals zero, while the 8$^{\text{th}}$ and 10$^{\text{th}}$ actuator effectiveness are respectively reduced by 53.20 % and 26.52 %. Equation (III.1.13) is used to model backlash, whereby $k_1 = k_2 = 1$, and

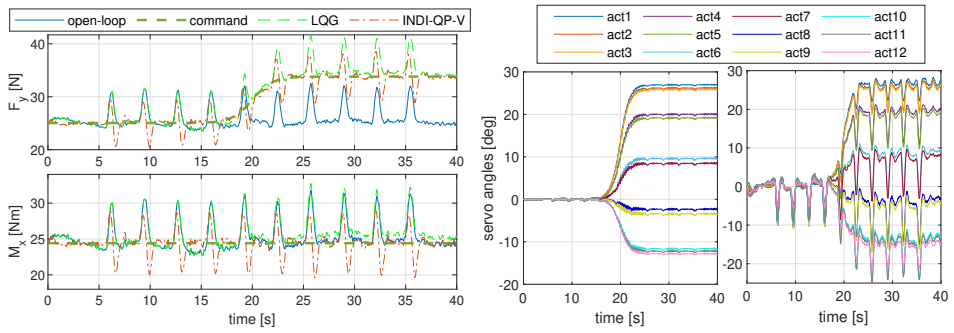


(a) Load responses.

(b) Servo angles: LQG (left) and INDI-QP-V (right).

Figure III.1.13: Simultaneous gust and manoeuvre load alleviation with measurement noise.

$u_{f_+} = -u_{f_-} = 0.6$ deg. Figure III.1.14 and Table III.1.2 show that the performance of LQG is further degraded by the fault and backlash. Although the reduction rate of RMS $(F_y - F_{y_*})$ is still positive under LQG control, the other performance metrics all become negative. On the contrary, INDI-QP-V can simultaneously alleviate gust and manoeuvre loads in spite of coloured noises, actuator fault, and backlash. The RMS value of $F_y - F_{y_*}$ is reduced by 68.08 %, which is very close to the experimental result (67.76 %). Under INDI-QP-V control, all the load metrics are reduced by over 19 % in the simulation and are alleviated by more than 44 % in the experiment.



(a) Load responses.

(b) Servo angles: LQG (left) and INDI-QP-V (right).

Figure III.1.14: Simultaneous gust and manoeuvre load alleviation with noise, fault, and backlash.

III.1.5. Conclusions

This Chapter presents the design and wind tunnel testing of a simultaneous gust and manoeuvre load alleviation control law for a seamless active morphing wing. To begin with, the Incremental Nonlinear Dynamic Inversion (INDI) control is derived for a generic multi-input/multi-output nonlinear system with an arbitrary relative degree. Then the closed-loop stability under the perturbation of model uncertainties, external disturbances, and

control allocation errors are analyzed using Lyapunov methods. Moreover, two control allocation methods and their corresponding stability criteria are derived for INDI control. Although the INDI with Pseudo Inverse Control Allocation (INDI-PI) provides the least-squares solution, the input constraints are not considered. On the contrary, the actuator position constraints, rate constraints, and relative position constraints can all be satisfied by INDI with Quadratic Programming Control Allocation (INDI-QP). Furthermore, INDI-QP is augmented with the virtual shape functions (denoted as INDI-QP-V) to ensure the smoothness of the morphing wing.

The effectiveness of the proposed INDI-QP-V has been validated by wind-tunnel experiments. During the experiment, the pick-up point of the 9^{th} actuator was broken; the morphing mechanisms also presented unexpected hysteresis backlash behaviours. Despite these challenges, experimental results show that INDI-QP-V is robust to aerodynamic uncertainties, gusts, actuator faults, and nonlinear backlash. In manoeuvre load alleviation tasks, INDI-QP-V increased the total lift for performing pull-up manoeuvres without amplifying the wing root bending moment. In the presence of successive "1-cos" gusts, INDI-QP-V mitigated the loads without requiring any gust information. Furthermore, INDI-QP-V made the seamless wing morph actively to modify the spanwise lift distribution and resist gusts at the same time. In all the tested cases, the input constraints were satisfied; the wing shape was smooth; the control law was realised in real-time.

To further demonstrate the features of INDI-QP-V, is has been compared to the Linear Quadratic Gaussian (LQG) control. As a linear control method, LQG has to be used along with the tedious gain-scheduling method for nonlinear control problems. Its robustness against model uncertainties and external disturbances is also not guaranteed. On the contrary, INDI-QP-V is a nonlinear control method with inherent robustness against uncertainties and disturbances. Moreover, INDI-QP-V has less model dependency, which simplifies its hardware implementation process. Furthermore, in simultaneous gust and manoeuvre load alleviation tasks, INDI-QP-V can more effectively alleviate excessive loads.

In conclusion, simulations and wind tunnel experiments have demonstrated that the proposed INDI-QP-V control is easy to implement, robust to actuator fault and backlash, and effective in simultaneously alleviating the gust and manoeuvre loads of the seamless active morphing wing. Scaling up the SmartX-Alpha wing to a full-scale flight aircraft will be explored in future work.

References

[1] X. Wang, T. Mkhoyan, I. Mkhoyan, and R. De Breuker, "Seamless Active Morphing Wing Simultaneous Gust and Maneuver Load Alleviation," *Journal of Guidance, Control, and Dynamics*, pp. 1649–1662, 2021.

[2] T. A. Weisshaar, "Morphing Aircraft Systems: Historical Perspectives and Future Challenges," *Journal of Aircraft*, vol. 50, pp. 337–353, Mar. 2013.

[3] J. E. Hubbard, "Dynamic Shape Control of a Morphing Airfoil Using Spatially Distributed Transducers," *Journal of Guidance, Control, and Dynamics*, vol. 29, pp. 612–616, may 2006.

[4] F. Previtali, A. F. A. Arrieta, and P. Ermanni, "Performance of a Three-Dimensional Morphing Wing and Comparison with a Conventional Wing," *AIAA Journal*, vol. 52, pp. 2101–2113, oct 2014.

[5] B. K. S. Woods, I. Dayyani, and M. I. Friswell, "Fluid/Structure-Interaction Analysis of the Fish-Bone-Active-Camber Morphing Concept," *Journal of Aircraft*, vol. 52, pp. 307–319, Jan. 2015.

[6] N. B. Cramer, D. W. Cellucci, O. B. Formoso, C. E. Gregg, B. E. Jenett, J. H. Kim, M. Lendraitis, S. S. Swei, G. T. Trinh, K. V. Trinh, and K. C. Cheung, "Elastic shape morphing of ultralight structures by programmable assembly," *Smart Materials and Structures*, vol. 28, p. 055006, apr 2019.

[7] T. Mkhoyan, R. N. Thakrar, R. De Breuker, and J. Sodja, "Design of a Smart Morphing Wing Using Integrated and Distributed Trailing-Edge Camber Morphing," in *Proceedings of the ASME 2020 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, (Irvine, USA), 2020.

[8] P. Fabre, X. Le Tron, and P. Lacoste, "System for reducing the forces applied to the wings and particularly to the root of the wings of an aircraft in flight," Feb. 16 1993. US Patent 5,186,416.

[9] M. d. F. V. Pereira, I. Kolmanovsky, C. E. Cesnik, and F. Vetrano, "Model Predictive Control Architectures for Maneuver Load Alleviation in Very Flexible Aircraft," in *AIAA Scitech 2019 Forum*, no. January, (San Diego, California), American Institute of Aeronautics and Astronautics, jan 2019.

[10] S. Haghighat, H. H. T. Liu, and J. R. R. A. Martins, "Model-Predictive Gust Load Alleviation Controller for a Highly Flexible Aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 6, pp. 1751–1766, 2012.

[11] E. Vartio, E. Shaw, and T. Vetter, "Gust Load Alleviation Flight Control System Design for a SensorCraft Vehicle," in *26th AIAA Applied Aerodynamics Conference*, no. August, (Honolulu, Hawaii), pp. 1–10, American Institute of Aeronautics and Astronautics, aug 2008.

[12] J.-W. van Wingerden, A. Hulskamp, T. Barlas, I. Houtzager, H. Bersee, G. van Kuik, and M. Verhaegen, "Two-Degree-of-Freedom Active Vibration Control of a Prototyped "Smart" Rotor," *IEEE Transactions on Control Systems Technology*, vol. 19, pp. 284–296, mar 2011.

[13] Y. Bi, C. Xie, C. An, and C. Yang, "Gust load alleviation wind tunnel tests of a large-aspect-ratio flexible wing with piezoelectric control," *Chinese Journal of Aeronautics*, no. December, 2016.

[14] D. H. Baldelli, D.-H. Lee, R. S. S. Pena, and B. Cannon, "Modeling and Control of an Aeroelastic Morphing Vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 31, pp. 1687–1699, nov 2008.

[15] N. T. Nguyen, E. Ting, D. Chaparro, M. C. Drew, and S. S.-M. Swei, "Multi-Objective Flight Control for Drag Minimization and Load Alleviation of High-Aspect Ratio Flexible Wing Aircraft," in *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, no. January, (Grapevine, Texas), American Institute of Aeronautics and Astronautics, jan 2017.

[16] "Multi-Objective Adaptive Control for Load Alleviation and Drag Minimization of Flexible Aircraft," (Kissimmee, Florida), American Institute of Aeronautics and Astronautics, jan 2018.

[17] S. J. Lee, K. E. Hashemi, M. C. Drew, N. T. Nguyen, and H. J. Kim, "Robust Gust Load Alleviation Control using Disturbance Observer for Generic Flexible Wing Aircraft in Cruising Condition," in *2018 Annual American Control Conference (ACC)*, vol. 2018-June, pp. 2257–2263, IEEE, jun 2018.

[18] X. Wang, E. van Kampen, Q. P. Chu, and R. De Breuker, "Flexible Aircraft Gust Load Alleviation with Incremental Nonlinear Dynamic Inversion," *Journal of Guidance, Control, and Dynamics*, vol. 42, pp. 1519–1536, jul 2019.

[19] R. Tekin, K. S. Erer, and F. Holzapfel, "Control of Impact Time with Increased Robustness via Feedback Linearization," *Journal of Guidance, Control, and Dynamics*, vol. 39, pp. 1682–1689, jul 2016.

[20] A. N. Kalliny, A. A. El-Badawy, and S. M. Elkhamisy, "Command-Filtered Integral Backstepping Control of Longitudinal Flapping-Wing Flight," *Journal of Guidance, Control, and Dynamics*, vol. 41, pp. 1556–1568, jul 2018.

[21] F. Grondman, G. Looye, R. O. Kuchar, Q. P. Chu, and E. van Kampen, "Design and Flight Testing of Incremental Nonlinear Dynamic Inversion-based Control Laws for a Passenger Aircraft," in *2018 AIAA Guidance, Navigation, and Control Conference*, no. January, (Kissimmee, Florida), American Institute of Aeronautics and Astronautics, jan 2018.

[22] S. Sun, X. Wang, Q. Chu, and C. C. de Visser, "Incremental Nonlinear Fault-Tolerant Control of a Quadrotor With Complete Loss of Two Opposing Rotors," *IEEE Transactions on Robotics*, pp. 1–15, 2020.

[23] X. Wang, E. van Kampen, Q. P. Chu, and P. Lu, "Incremental Sliding-Mode Fault-Tolerant Flight Control," *Journal of Guidance, Control, and Dynamics*, vol. 42, pp. 244–259, feb 2019.

[24] I. Matamoros and C. C. de Visser, "Incremental Nonlinear Control Allocation for a Tailless Aircraft with Innovative Control Effectors," in *2018 AIAA Guidance, Navigation, and Control Conference*, no. January, (Kissimmee, Florida), pp. 1–25, American Institute of Aeronautics and Astronautics, jan 2018.

[25] A. L. Fradkov, I. V. Miroshnik, and V. O. Nikiforov, *Nonlinear and Adaptive Control of Complex Systems.* Dordrecht: Springer Netherlands, vol. 491 ed., 1999.

III.1

[26] H. K. Khalil, *Nonlinear Systems*. New Jersey: Prentice-Hall, 2002.

[27] X. Wang, E. van Kampen, Q. Chu, and P. Lu, "Stability Analysis for Incremental Nonlinear Dynamic Inversion Control," *Journal of Guidance, Control, and Dynamics*, vol. 42, pp. 1116–1129, may 2019.

[28] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[29] R. Bartlett, A. Wachter, and L. Biegler, "Active set vs. interior point strategies for model predictive control," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, vol. 6, pp. 4229–4233 vol.6, IEEE, 2000.

[30] J. C. Mason and D. C. Handscomb, *Chebyshev polynomials*. Boca Raton, Florida: CRC press, 2002.

[31] M. M. Gomroki, F. Topputo, F. Bernelli-Zazzera, and O. Tekinalp, "Solving Constrained Optimal Control Problems Using State-Dependent Factorization and Chebyshev Polynomials," *Journal of Guidance, Control, and Dynamics*, vol. 41, pp. 618–631, mar 2018.

[32] Y. Ferrier, N. T. Nguyen, E. Ting, D. Chaparro, X. Wang, C. C. de Visser, and Q. P. Chu, "Active Gust Load Alleviation of High-Aspect Ratio Flexible Wing Aircraft," in *2018 AIAA Guidance, Navigation, and Control Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, Jan. 2018.

[33] T. Theodorsen, "General Theory of Aerodynamic Instability and the Mechanism of Flutter," tech. rep., NACA Technical Report No. 496 (NACA-TR-496), 1935.

[34] OptiTrack, "Motive API NaturalPoint Product Documentation Ver 2.2."

[35] P. Gold and M. Karpel, "Reduced-size aeroservoelastic modeling and limit-cycle-oscillation simulations with structurally nonlinear actuators," *Journal of Aircraft*, vol. 45, no. 2, pp. 471–477, 2008.

[36] K. D. Frampton and R. L. Clark, "Experiments on control of limit-cycle oscillations in a typical section," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 5, pp. 956–960, 2000.

[37] J. Campos, F. L. Lewis, and R. Selmic, "Backlash compensation in discrete time nonlinear systems using dynamic inversion by neutral networks," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2, no. 2, pp. 1289–1295, 2000.

[38] P. Blelloch and D. Mingori, "Robust linear quadratic gaussian control for flexible structures," *Journal of Guidance Control and Dynamics*, vol. 13, pp. 66–72, 1990.

[39] X. Wang, T. Mkhoyan, and R. De Breuker, "Nonlinear incremental control for flexible aircraft trajectory tracking and load alleviation," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 1, pp. 39–57, 2022.

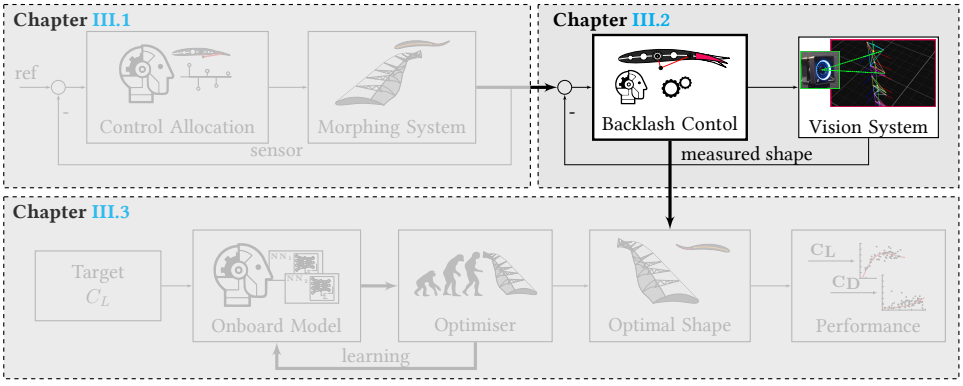
[40] S. Waitman and A. Marcos, "H∞ control design for active flutter suppression of flexible-wing unmanned aerial vehicle demonstrator," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 4, pp. 656–672, 2020.

[41] M. Green and D. J. Limebeer, *Linear robust control.* Courier Corporation, 2012.

III.2

Mitigating Backlash and Disturbances Using Vision-Based Control

In Part II and the previous Chapter, mechanical imperfections were identified, and it was established that backlash hysteresis, in particular, contributed to nonlinearities in the actuation. In this Chapter, a novel vision-based incremental control approach is introduced to mitigate mechanical imperfections. An experiment is performed, demonstrating that the compensating controller can follow the tracked command well without system design modifications [1].



This Chapter is based on the following journal paper:
B. Sun, T. Mkhoyan, E.-J. Van Kampen, R. De Breuker, and X. Wang, "Vision-Based Nonlinear Incremental Control for A Morphing Wing with Mechanical Imperfections," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–13, 2022

III.2

With the increasing desire of the aerospace industry to reduce emissions and fuel consumption, morphing wings have gained much interest due to the ability to adapt the wing shape in-flight for improved efficiency and fuel consumption. A smart morphing wing, the SmartX-Alpha, was designed, manufactured, and tested in the previous chapters. In Chapter III.1 the SmartX-Alpha demonstrated how wing load could be alleviated while achieving the optimal lift distribution. However, the widely existing mechanical imperfections can degrade the performance of the morphing wing and even lead to instabilities. This Chapter proposes a vision-based adaptive control approach to compensate for mechanical imperfections and tackle these issues actively. In this approach, an incremental model is constructed online to identify the system dynamics using servo commands and vision measurements. Then nonlinear dynamic inversion control is applied based on the identified model. Real-world experiments on the SmartX-Alpha have validated this data-driven control approach with visual feedback. The results demonstrate that the vision-based system combined with the proposed control methodology can actively compensate for mechanical imperfections with minimal adjustments to the actual system design. Compared to a controller that only uses a feed-forward input/output mapping, this proposed approach improves the system performance and decreases the tracking errors by more than 62 % despite disturbances. The results collectively demonstrate the effectiveness of the proposed control system, which sets a foundation for realising morphing in next-generation aircraft.

III.2.1. Introduction

Active morphing can bring several benefits to conventional wing designs. Morphing wings have the potential to improve aircraft performance across the entire flight envelope by actively adapting the shape. Due to conflicting requirements [2], conventional wing designs generally can only be optimised for one single flight condition, such as cruise. The SmartX project [3] was initiated at the Delft University of Technology to assess the benefits of morphing wings.

In Part II, an over-actuated and over-sensed wing prototype was developed for this project, named SmartX-Alpha, capable of seamless active wing morphing with six distributed Translation Induced Camber (TRIC) morphing modules [4]. Coupled with advanced nonlinear control methods, this wing has demonstrated the capability to actively reduce gust loads while actively maintaining an optimal lift distribution in the wind tunnel study presented in Chapter III.1 [5]. However, due to the nature of the morphing mechanism, the mechanical complexity, and the manufacturing imperfections, the wing design exhibited hysteresis nonlinearities in the actuation. The nonlinear backlash phenomenon was observed in both wind tunnel tests and a design validation assessment with a digital Digital Image Correlation (DIC) setup, shown in Chapter II.1. This backlash effect diminishes the achievable morphing range and consequently reduces the aerodynamic control effectiveness of the wing.

Apart from backlash, other forms of mechanical imperfections also exist in aerospace systems, such as input saturation [6], friction [7], deadzone [8], etc. These mechanical imperfections can largely degrade the system performance and lead to undesirable phenomena such as limit-cycle oscillations, divergence, and even flutter [8, 9]. Instead of perfecting hardware, which is costly and non-adaptative, an alternative is to compensate for these hardware imperfections via software algorithms actively.

Backlash compensation is the most challenging of all the aforementioned mechanical imperfections because of its time sequence dependency. Some researchers attempt to compensate for backlash in the simulation environment via an inversion approach with

general approximators (e.g., neural networks) [10], and fuzzy controllers [11]. The backlash dynamics are assumed to be known or unknown but invariant in these existing publications, and global regulation is adopted for inversion. However, in a real system such as the SmartX-Alpha wing, owing to uncertainties and disturbances, the backlash nonlinearity is not always known and invariant, which leads to invalidation of the global model-based inversion method. The limitations in the existing methods motivate our research in investigating online active compensation methods with no need for known or invariant dynamics.

To actively compensate for mechanical imperfections, online identification techniques are necessary such that the controller can adapt to constantly changing light conditions and environments. Commonly used grey-box identification approaches [12] require full state feedback and are therefore not applicable for a system such as the SmartX-Alpha. In such systems, uncertainties and nonlinearities in the system are present and cannot be made available through internal states.

Identification methods which are output-only are, in contrast, a suitable choice for this system. In a recent study, an effective online identification method for dynamic systems was proposed, which requires only output-feedback (OPFB) and less computation power than neural networks in [13]. This method identifies the so-called incremental [14] model by utilising interval-based linearisation and discretisation using sensor feedback. Combined with intelligent control methods, the incremental model has successfully been applied to various aerospace systems [13, 15, 16]. However, the effectiveness of the incremental model in the real-world system has not been validated yet.

For the controller to adapt to changes in the system dynamics, not only an online identification approach is required but also an adaptive control law. A conventional gain-scheduling approach using linear control methods is insufficient since stability and performance are not guaranteed between operational points [17]. The Non-linear Dynamic Inversion (NDI) is a widely used nonlinear control method, widely applied in the aerospace community [12, 17–19], which allows cancelling the nonlinearities in a nonlinear system such that the closed-loop dynamics is linear [12, 17]. For mitigation of backlash in the current system, which exhibits a high degree of uncertainty, the Non-linear Dynamic Inversion (NDI) paired with the incremental model is believed to be a suitable approach.

To enable real-time mechanical imperfection compensation in the real world, the controller has to know the shape of the morphing wing in real-time. Furthermore, our study aims to provide a solution for an existing imperfect mechanical design, preferably imposing minimal adjustments to the hardware. In this study, we propose to accomplish this task by vision-based control. In our previous research, vision-based tracking has demonstrated its effectiveness in flexible and morphing system shape estimations, featured by its non-invasive and model-free nature [20]. Coupled with nonlinear filtering techniques, a vision-based tracking system can be retro-fitted to an existing wing design and provide the capability to measure wing vibrations in the presence of gusts [20]. This setup can also be used as a stabilisation system for flapping wings [21], obstacle avoidance for UAVs [22], and flight manoeuvring tracking [23]. Vision-based control can also benefit truss-based morphing mechanisms with many actuators [24], which is an active research field in robotics [25, 26].

The main contributions of this Chapter are summarised as follows.

- Experimental verification of a vision-based control method using Incremental Model-based Nonlinear Dynamic Inversion (IM-NDI) with online identification performed on the SmartX-Alpha wing.
- The methodology and real-world implementation techniques for real-time vision-based robust morphing wing shape reconstruction using distributed infrared sensors are presented.
- The mechanical imperfections (including backlash, friction, and hysteresis) in the morphing wing are actively compensated for by the vision-based IM-NDI approach without invasive modification to the system.

The remainder of this Chapter is organised as follows: Sec. III.2.2 introduces the Incremental Model-based Nonlinear Dynamic Inversion control approach. Section III.2.3 describes the development of the vision-based control system. The experiment study is presented in Sec. III.2.4, and Sec. III.2.5 summarises the work of this Chapter.

III.2.2. Incremental Model-Based Nonlinear Dynamic Inversion Control

This section presents a brief overview of the controller design using the dynamic inversion control method and online model identification approach. The layout of the approach is illustrated in Fig. III.2.1. The identification approach relies on an incremental data-driven model, proposed in [13], which can be identified online using Recursive Least Squares (RLS) algorithm and vision-based sensor data. The control design specifics, incremental model derivation, and proofs are presented in [1].

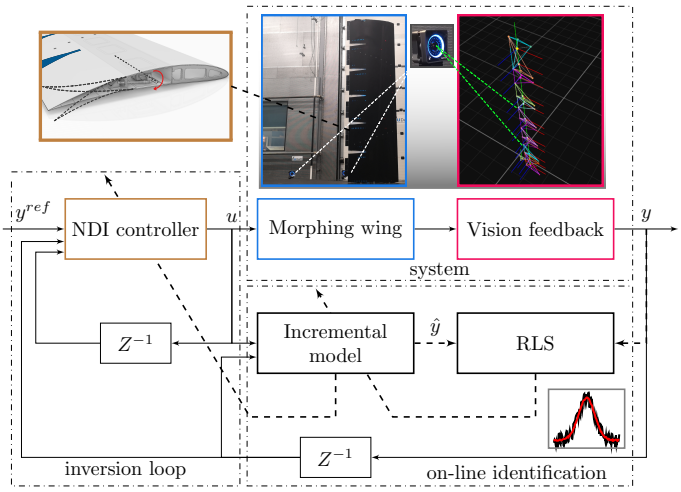


Figure III.2.1: The architecture of the IM-NDI controller, where the solid lines denote the signal flows and the dashed lines represent the parameter update paths.

The morphing wing can be represented by the following nonlinear multi-input multi-output system:

$$\dot{x}(t) = f(x(t), u(t)) + d(t),\ y(t) = h(x(t)) \tag{III.2.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $y \in \mathbb{R}^p$ are the system state vector, control input vector, and measurable output state vector, respectively. $f : (\mathbb{R}^n,\ \mathbb{R}^m) \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are assumed Lipschitz continuous on their domains. The nonlinear system is assumed to be both controllable and observable.

To ensure that the controller constantly adapts to changes in the system and environment, online system identification, based on incremental sensory feedback, is performed. The so-called incremental model, proposed in [13], is constructed using linearisation and discretisation techniques. A fundamental assumption of this approach is that the sampling frequency is sufficiently high compared to the time-variation of the system dynamics. This ensures that errors caused by linearisation and discretisation can be bounded within a small vicinity of zero within the sampling interval Δt [1]. In previous studies, this assumption has been shown to hold for a sampling frequency of 100 Hz [17, 27]. Experimental results presented in Chapter III.1 demonstrated that a sampling frequency of 60 Hz was sufficient for morphing wing control.

Given the reference input signal and the system output, as shown in Fig. III.2.1, the tracking error, $e = y - y^{\text{REF}}$, is defined as the difference between the two signals. The tracking error can be discretised to represent the error increment, Δe_{t+1}, considering the stable tracking error dynamics of the system given by Eq. (III.2.1).

Theorem 5. *[13] Given that the nonlinear system defined by Eq. (III.2.1) is observable, then under sufficiently high sampling frequency, the output tracking error increment* $\Delta e_{t+1} = \Delta y_{t+1} - \Delta y_{ref,t+1}$ *can be determined uniquely from the observations and control inputs over a sufficiently long time horizon,* $[t - M + 1, t],\ M \geq (n + p)/p$*:*

$$\Delta e_{t+1} \approx F_t \overline{\Delta e}_{t,M} + G_t \overline{\Delta u}_{t,M}, \tag{III.2.2}$$

where $F_t \in \mathbb{R}^{p \times Mp}$ is the identified transition matrix, and $G_t \in \mathbb{R}^{p \times Mm}$ is the identified input distribution matrix. Furthermore, $\overline{\Delta u}_{t,M} = [\Delta u_t^{\mathrm{T}}, \Delta u_{t-1}^{\mathrm{T}}, ..., \Delta u_{t-N+1}^{\mathrm{T}}] \in \mathbb{R}^{Nm}$ and $\overline{\Delta e}_{t,M} = [\Delta e_t^{\mathrm{T}}, \Delta e_{t-1}^{\mathrm{T}}, ..., \Delta e_{t-N+1}^{\mathrm{T}}] \in \mathbb{R}^{Np}$ are the input and tracking error data of N previous samples, respectively.

For slow varying dynamics, $F_t \overline{\Delta e}_{t,M} + G_t \overline{\Delta u}_{t,M} - y_{\text{ref},t} \approx -k_p \Delta t e_t - k_d \Delta e_t$ can be approximated, where $k_p > 0$, $k_d > 0$ are control parameters [1]. From Eq. (III.2.2), the nonlinear dynamic inversion control in its discrete form can be applied, which regulates the tracking error to zero asymptotically despite model uncertainties, external disturbances, and backlash.

III.2.3. Vision-Based Control Approach

A crucial aspect of implementing a control strategy to compensate for mechanical imperfections is an accurate knowledge of the morphing wings' shape. In particular, the variable of interest to the controller is the local vertical displacement of the wing trailing edge concerning a body-fixed coordinate system. In a previous study, a morphing wing concept utilising the distributed translation induced camber (TRIC) has been described [4]. This design has a relatively stiff wing box and a flexible morphing trailing edge.

A body-fixed coordinate system, $\mathcal{F}_B$ is chosen to be near the root of the wing in the wing box section, with an origin O_B. The displacement of the trailing edge, denoted as $\boldsymbol{z} = [z_1, z_2, ..., z_{12}]^{\mathrm{T}}$ along 12 stations of the span is reconstructed in the $\mathcal{F}_B$ frame, from a camera-fixed frame $\mathcal{F}_C$ in real-time, utilising vision-based tracking. Two locations in each of the six modules are tracked and fed back to the controller. The experimental setup is shown in Fig. III.2.2.

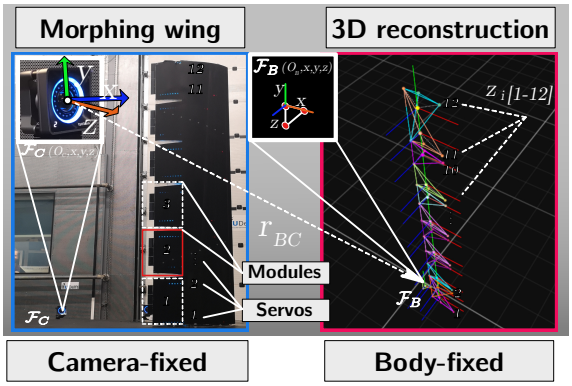


Figure III.2.2: The experimental apparatus with vision-based control components.

III.2.3.1. Apparatus

The experimental apparatus is shown in Fig. III.2.2. The system consists of a morphing wing with six distributed TRIC morphing modules placed in the Open Jet Facility (OJF) wind tunnel facility of the Delft University of Technology. Each module is actuated by two embedded servos [4]. An array of infra-red (IR) light-emitting diodes (LEDs) of type 3528 850NM WLP PLCC2 [28], emitting IR in the 850 nm wavelength, are installed on the wing bottom surface in the non-morphing wing-box and the morphing trailing-edge modules. The array of IR led is powered by a 12 V DC power supply, and the brightness is actively controlled by IRF520 Power metal–oxide–semiconductor field-effect transistor (MOSFET) dimmer circuit [29]. Four IR markers per module (24 in total) with another three markers defining the body-fixed reference frame are tracked by five Prime^{x}41 4.1 megapixel IR cameras at a frame rate of 250 frames per second (FPS) [30]. The deflections of the morphing flaps are reconstructed in real-time with a reconstruction algorithm. The markers' coordinate system definitions and layout are explained in Sec. III.2.2.Each morphing module is tracked by four reference IR markers, where two are installed in the non-morphing wing box directly behind the cut and the remainder in the trailing edge. The IR markers are arranged so that all corners of the module can be tracked in camber and twist motion. Another three IR LED units are installed in the wing root at right angles in a triangular configuration, representing the body-fixed 3-axis coordinate system.

The intensity of IR has been adjusted to approximately 20 percent and tuned to obtain the best tracking performance. Image segmentation and filtering are applied to improve further the tracking, which will be discussed in Sec. III.2.3.2.

III.2.3.2. Processing Framework

A 3D reconstruction procedure is required to transform the measured marker correspondence $x_1, x_2, \ldots, x_n$ detected in the image frame, u, v, to a 3D world coordinates defined in the camera fixed reference frame, $\mathcal{F}_C$. Segmentation and filtering are applied to the raw images to obtain the binary mask with marker locations in the image frames for each camera. The filtering consists of image threshold filtering and morphological image transformations [31] to improve the segmentation of distinct LED markers. An example of similar segmentation and filtering approach is presented in [20]. Dynamic adjustment of the IR brightness was needed to prevent two or more markers from merging into a single blob for far-away camera views. A concise overview of the vision-based tracking pipeline is shown in Fig. III.2.3 and explained in the following subsections.

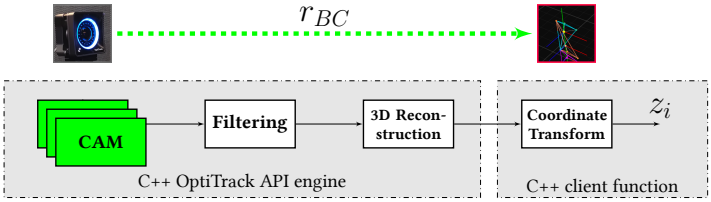


Figure III.2.3: A concise vision-based tracking pipeline.

In a previous study in OJF, a 3D reconstruction approach with a stereo-camera setup was developed, which showed camera calibration sensitivity due to the wind tunnel's adverse environmental conditions (flow conditions and mechanical vibrations) [20]. A generally suitable approach to improve the tracking accuracy and add redundancy to the tracking system is a multi-camera (> 2) setup [32]. Therefore, a five-camera setup was used in this study to improve robustness against calibration drift developed over time.

The problem of 3D reconstruction with two or more camera setups can be performed with several triangulation methods. The general triangulation problem in a two-view setup tries to find the best estimate of point in 3D denoted with vector X given its projections x_1, x_2 (point correspondence) in two camera views and camera projection matrices P_1, P_2, defining the location, orientation (extrinsic parameters) and the properties of the camera lens (intrinsic parameters) of each camera. In the presence of measurements errors, finding the best estimate of 3D point $\hat{X}$ from noisy correspondence $\bar{x_1}, \bar{x_2}$ is treated at a minimisation problem denoted by triangulation function τ: $X = \tau(x_1, x_2, P_1, P_2)$. A common triangulation method used for linear perspective transformation is a Direct Linear Transform (DLT) which solves the optimisation problem by sSingular Value Decomposition (SVD) [33].

The n-view 3D reconstruction problem is concerned with finding the optimal estimation of an object $\hat{X}$ in a 3D global coordinate frame (i.e., locations in the x, y and z-axes), which is observable in noisy $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$ points correspondence in n camera views. The point correspondences $\bar{x}_i$ are generally defined by markers in u, v coordinates of a 2D image plane. Back-projecting the 3D point onto the respective camera views, a minimisation problem can be defined to find the re-projection error, $E = \sum_{k=1}^{n} ||x_k - \bar{x}_k||^2$, and solved by an expanded linear system of equations similar to SVD in a DLT procedure [34]. Global optimisation methods can be applied, such as algebraic, matrix inequality and the

$\mathcal{L}_\infty$ approach [35]. The development of a particular n-view triangulation method is not considered in this study and was already presented in the earlier work conducted in the scope of this dissertation [36].

To perform the 3D point cloud reconstruction in real-time, a proprietary reconstruction engine is used by OptiTrack API [37]. The reconstructed coordinates are streamed via NatNet [38] client/server application over the Ethernet protocol; then, coordinate transformations are applied to obtained flap position in body-fixed coordinates. A decentralised control architecture, capable of 1 kHz synchronisation, synchronises the data and real-time [39]. All the applications for processing, reconstructing and accessing the data are written in low-level C++ programming language for best performance. Multi-camera calibration is performed by wandering process, resulting in an average calibration error of 0.25 mm for all cameras. The accuracy of a similar setup has been verified in [40].

The final step in the vison-based tracking pipeline is a coordinate system transformation from the global camera-fixed coordinate system, $\mathcal{F}_C$ with an origin O_C, to the body body-fixed coordinate system, $\mathcal{F}_B$ with an origin O_B. This transformation is needed to express the relative deflections of the trailing-edge modules z_i concerning the baseline un-morphed shape. The coordinate frames and their respective origins, located approximately 2 meters away, are connected by a vector r_{BC} as shown in Figure III.2.2. The transformation $\mathcal{F}_C \rightarrow \mathcal{F}_B$ is achieved by a translation, followed by 3-axis rotations in pitch, roll, and yaw axes (θ, ϕ, ψ). The transformations are performed continuously as the morphing may continuously exhibit motions relative to the frame $\mathcal{F}_B$. The average total processing latency was in the range of 5-7 ms, which is smaller than the control sampling interval (16.67 ms).

Four tracking markers define each trailing-edge morphing module; two are attached to the non-morphing wing-box, and two to the morphing trailing edge. In each morphing module, two tracking bodies are defined, responsible for the deflection estimate of, e.g., z_1 and z_2 for the module, along the module span in F_B frame. Two markers within each module are shared, as shown in (Fig. III.2.2). In total, 24 marker points are defined, with three additional points for the definition of the body's fixed origin located at the root of the wing.

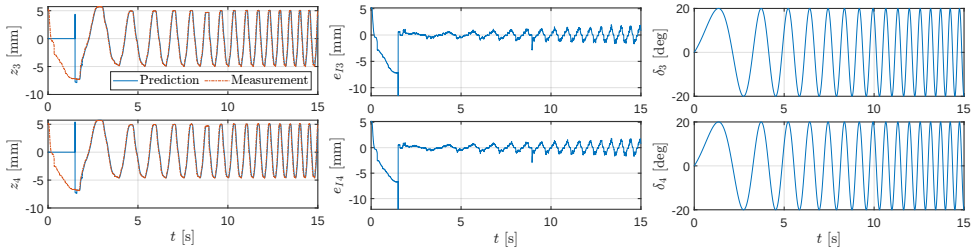
III.2.4. Experimental Results and Discussions

In the following sections, the effectiveness of the proposed control approach toward the backlash effect is verified in an experimental study with the morphing wing system. As described in Section III.2.3, the system consists of 6 modules, each driven by two embedded actuators [4]. Module 2 is chosen for validation, and its adjacent modules are used to produce disturbances, as all modules exhibit comparable performance, and the disturbance is unknown. The two control channels of Module 2 are indexed by subscripts 3 and 4, respectively. It is noted that these two channels are identified together, leading to a 2-input-2-output system. The control command and vision feedback data are transmitted at 60 Hz between the host computer and the physical system, while the identifier and the controller work in a host computer at 500 Hz.

Online Identification Performance

Prior to evaluating the closed-loop control performance, the online identification performance of the incremental model is evaluated. The width of the sliding window is set as $N = 50$, which means 50 previous data sets stored in the host computer are utilised, rather than 50 real samples. The forgetting factor γ_{RLS} is set to be 0.99995 such that the more recent data set has more dominant weight. $\underline{F}_t$, $\underline{G}_t$ and $\underline{\text{Cov}}_t$ are initialised as $\underline{F}_0 = [I_2, 0_{2\times 98}]$, $\underline{G}_0 = [I_2, 0_{2\times 98}]$ and $\underline{\text{Cov}}_0 = 10^3 \cdot I_{200}$, respectively, where $0_{2\times 98}$ denotes the 2-row-98-column matrix with all zero elements.

The identification effectiveness of the incremental model is validated in an open-loop manner using the sinusoidal control input signal, with an amplitude of $A_\delta = 20$ deg, and an angular frequency ω_δ sweeps from 0.2π rad/s to 4π rad/s. The identification is activated 1.5 s after the open-loop control process begins. As illustrated in Fig. III.2.4, the predicted displacements converge quickly to their values measured by the vision system as the identification is activated. The identification errors reach the minimum values at around $t = 3$ s, and then keep increasing as the angular frequency increases. Overall, despite some disturbances and outliers, the identification errors, which are mainly caused by delays, can be bounded within ± 2 deg. The experiment results verify the incremental model's effectiveness, making it suitable for closed-loop control purposes.



(a) Identification performance (b) Identification error (c) Control input

Figure III.2.4: Open-loop identification performance with a varying-frequency sinusoidal control signal.

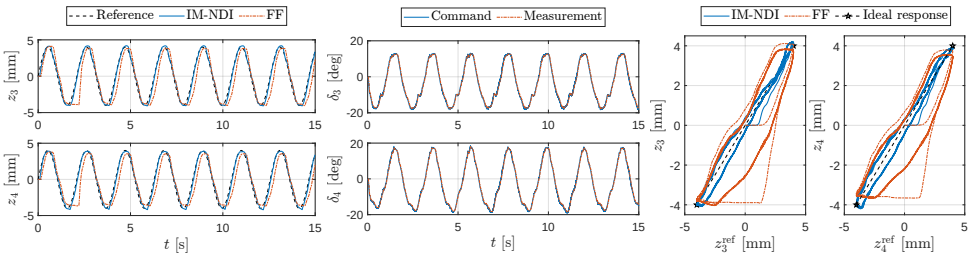
Closed-Loop Control Performance

The closed-loop control performance of the Incremental Model-based Nonlinear Dynamic Inversion (IM-NDI) is evaluated and compared to the Feed-Forward (FF) control method. A 2×2 FF mapping matrix between the servo angular inputs and the corresponding trailing-edge displacements is identified from the physical system. This matrix directly converts the morphing displacement commands to the servo commands in the FF control cases. For IM-NDI, we experimentally choose $k_p = 22.5$ and $k_d = 1.5$. These gains are tuned considering the trade-off between tracking error reduction and noise attenuation. The experiments are conducted to track a sinusoidal signal with the same amplitude $A_m = 4$ mm at varying angular frequencies, ω in range [3 rad/s-12 rad/s] and step 3 rad/s steps. For conciseness, the results for$\omega = 3$ rad/s and $\omega = 12$ rad/s are selected as representatives, shown in Figs. III.2.5 and III.2.6. These figures show that both IM-NDI and FF can successfully track the reference, but IM-NDI has smaller lags and reduced tracking errors.

In Figs. III.2.5 and III.2.6, the control command is the direct output of the IM-NDI controller, while the "measurement" denotes the real angle feedback from the servo. Although small oscillations occur, the actuator shows desirable performance for command tracking. Coping with mechanical imperfections is one of the most challenging issues in this task. Backlash is the most dominant and influential nonlinearity in this system among all mechanical imperfections, which is also the primary cause of the tracking lag. As illustrated in subfigure (c), the ideal tracking curve is a line segment defined on $[-4, 4]$ mm with a slope of 1. Due to the backlash, the sinusoidal reference and the actual measurement are characterised by a circle curve. It is clear that IM-NDI outperforms the FF controller in handling backlash nonlinearity because the curve of IM-NDI is closer to the ideal tracking line.

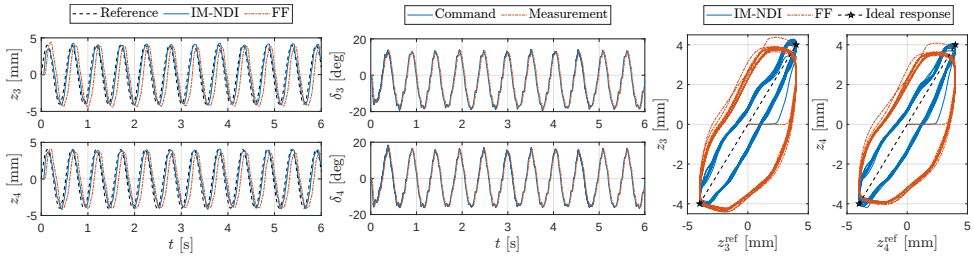
Figure III.2.7 represents an illustrative example of the performance metrics defined to intuitively compare the tracking performance in terms of the width and length of the backlash circle. Here, the left plot illustrates the standard backlash nonlinearity with a width of 2 mm. The ideal tracking is defined as a black dashed line whose endpoints are marked as stars. The width W_b and length L_b, are the origin-centred horizontal line segment spanning the with and the portion of the ideal tracking segment, spanning the length of the backlash.

Generally, the width W_b is the more common metric for describing the backlash; however, as revealed by the actual backlash curve, shown in the right plot of Fig. III.2.7, the real-world nonlinearity is complex and does not precisely obey the mathematical representation of the standard backlash in [10]. Therefore, both W_b and L_b are used for assessment in this Chapter. The width measurement W_b intends to describe the lagging property when changing the command direction, and a smaller W_b represents better performance. Moreover, the length measurement L_b can reflect the magnitude shrinking effect, and a larger L_b indicates better performance. Figure III.2.7 shows that as compared to the standard backlash nonlinearity, the effect caused by the real-world backlash nonlinearity is mainly reflected on the W_b, whereas the magnitude shrinking phenomenon is less severe. The control performance comparison regarding different angular frequencies is summarised in Table III.2.1, and the data represent the average value of the two actuation channels. It can be observed that thanks to the active compensation, IM-NDI outperforms FF in both W_b and L_b for all angular frequencies, and even the worst case of IM-NDI is better than the best one of FF.



(a) Tracking performance (b) Control input of IM-NDI (c) Backlash nonlinearity

Figure III.2.5: Control performance when tracking a sinusoidal signal with $\omega = 3$ rad/s, $A_m = 4$ mm.



(a) Tracking performance (b) Control input of IM-NDI (c) Backlash nonlinearity

Figure III.2.6: Control performance when tracking a sinusoidal signal with $\omega = 12$ rad/s, $A_m = 4$ mm.

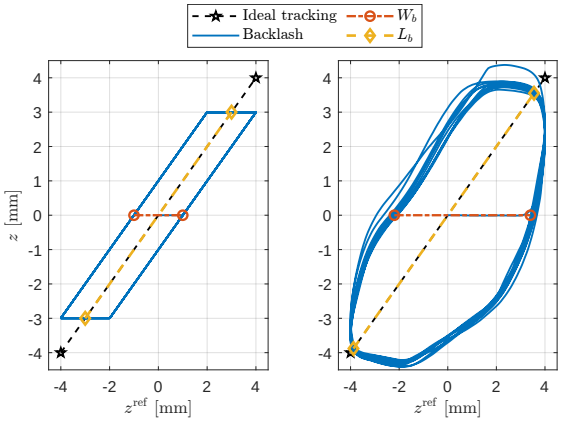


Figure III.2.7: Illustrative example of the performance metrics for the backlash nonlinearity.

Table III.2.1: Performance comparison regarding the backlash compensation, [mm]

	ω [rad/s]	3	6	9	12	Ideal
W_b	FF	2.5050	3.3850	4.5000	5.6000	0
	IM-NDI	0.6250	1.3550	1.9750	2.0750	0
L_b	FF	10.0409	10.1399	10.1823	10.1116	11.3137
	IM-NDI	11.0592	11.0309	10.9743	10.8187	11.3137

Closed-Loop Control Performance with Disturbance

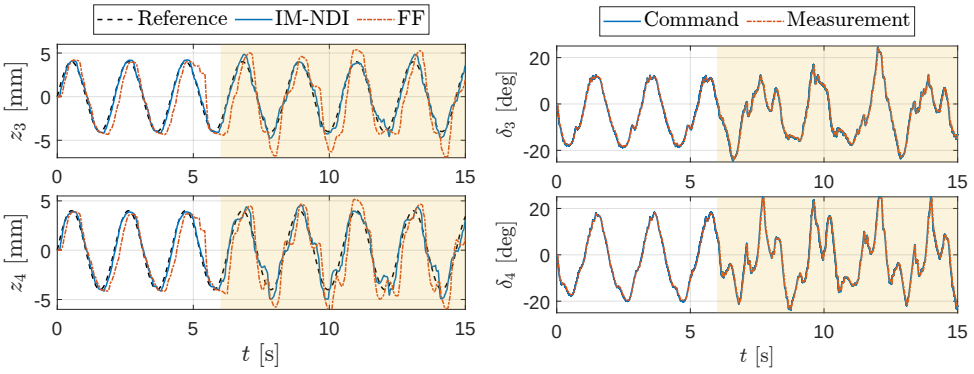
The last part of the results contains the effect of external disturbances on the robustness of the approach. As presented in Chapter II.1 the actuation of the existence of the elastomeric skin impacts the distributed modules of the SmartX-Alpha, smooths the airflow, and limits the drag.

The disturbance is injected as interfering actuation from modules 1 and 3 in the form of an open-loop sinusoidal signal, while the setting of module 2 is kept identical. The directive thus makes the controller's task more challenging to follow the reference while mitigating both the disturbance and backlash. For the disturbance generated by module 1, an open-loop sinusoidal signal of amplitude $A_d = 20$ deg, angular frequency of $\omega_d = 5$ rad/s and a zero phase $\phi_d = 0$ is chosen. For module 2, the disturbance signal parameters are chosen

as $A_d = 25$ deg, $\omega_d = 10$ rad/s and $\phi_d = \frac{\pi}{2}$ rad , respectively. The tracking performance is shown in Figs. III.2.8-III.2.9, where the shadowed area stands for the period that the external disturbances are injected.

It must be noted that the disturbances are proportionally significant; hence, the control performance of both methods is expected to degrade to certain extents. For the FF controller, the inability to adapt to change in actuation dynamics is much more significant, as seen from Figs. III.2.8-III.2.9. The IM-NDI, in contrast, manages to track the given reference despite disturbances by adaptively adjusting the control inputs. This is confirmed by the root-mean-square (RMS) of the tracking errors shown in Table III.2.2. Here, the RMS values are averaged over the actuator pair signal driving module two for the remaining angular frequencies in the range (3 and 12 rad/s). While it is observed that the tracking errors are higher for increasing angular frequencies, the IM-NDI approach can constrain the RMS within 0.9 deg.

For completeness, tracking errors are also represented in boxplots of Fig. III.2.10. The trend of higher tracking error with an increased angular frequency of the reference signal is confirmed for both methods, which is believed to be a direct consequence of physical limitations of the servo bandwidth (i.e. faster servo would be more effective at tracking in both cases). However, the FF error shows skewness in standard deviation due to the inability to adapt to disturbances. IM-NDI error, in contrast, appears to be more predictable and has zero mean.



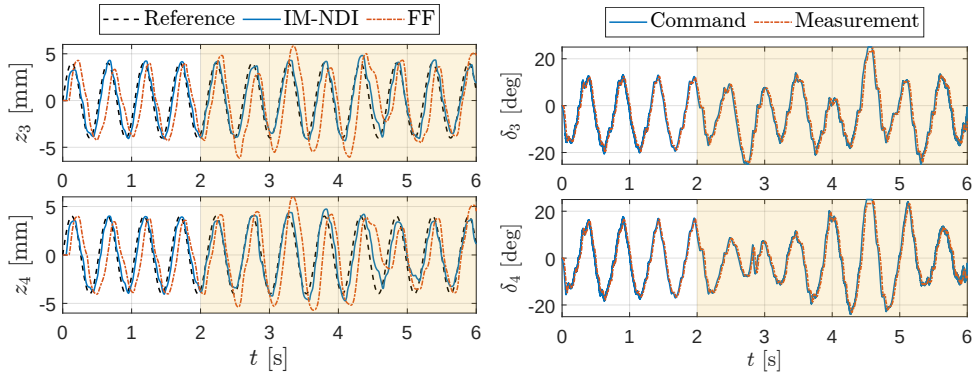
(a) Tracking performance

(b) Control input of IM-NDI

Figure III.2.8: Disturbance rejection performance when tracking a sinusoidal signal with $\omega = 3$ rad/s, $A_m = 4$ mm. The shaded area denotes the disturbance injection phase.

Table III.2.2: Comparison of the RMS of tracking errors, [deg]

ω [rad/s]	3		6		9		12	
Disturbed	No	Yes	No	Yes	No	Yes	No	Yes
FF	1.17	1.76	1.55	2.19	1.95	2.43	2.37	2.98
IM-NDI	0.34	0.62	0.55	0.74	0.73	0.86	0.84	0.87
Improvement	71.06%	64.78%	64.40%	66.34%	62.61%	64.69%	64.69%	70.82%



(a) Tracking performance

(b) Control input of IM-NDI

Figure III.2.9: Disturbance rejection performance when tracking a sinusoidal signal with $\omega = 12$ rad/s, $A_m = 4$ mm. The shaded area denotes the disturbance injection phase.

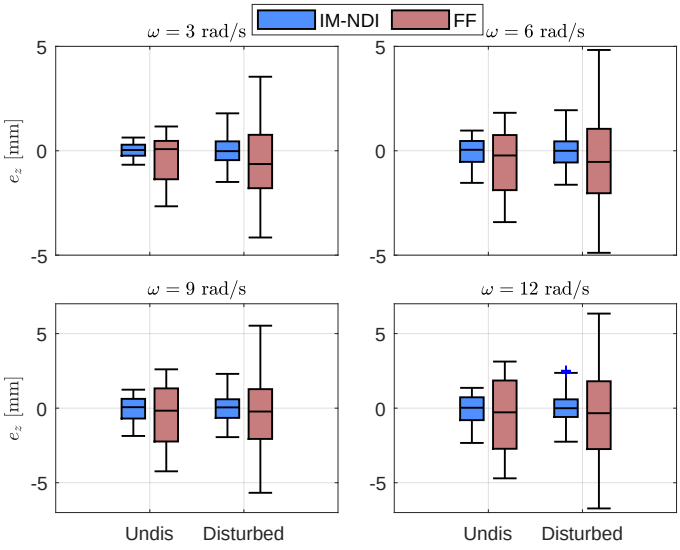


Figure III.2.10: Distribution of tracking errors under disturbed and undisturbed conditions.

III.2.5. Conclusions

This Chapter presents the experimental verification of a data-driven Incremental Model-based Nonlinear Dynamic Inversion (IM-NDI) control approach designed to mitigate the mechanical imperfections in a seamless active morphing wing. The system dynamics are identified online using the stored input/output data without a prior-known model. Then the NDI controller is developed based on the identified dynamics.

A crucial aspect of improving these imperfections is an accurate knowledge of the morphing wing's shape. A vision-based control system was developed, which has shown to be adequately effective for this task, given its robustness, high frame rates (250 FPS),

and good calibration accuracy (average 0.25mm). A real-world experiment is conducted based on computer vision feedback to evaluate the proposed method. The experimental results demonstrate that the morphing wing can track reference signals with different frequencies despite external disturbances by applying the IM-NDI. Under feed-forward control, the morphing wing suffers from mechanical imperfections, reflected by the lagging and magnitude shrinking phenomena in the tracking responses. The performance of feed-forward control also degrades in the presence of external disturbances. By contrast, experimental results show that IM-NDI can effectively decrease the tracking errors by more than 62 % despite disturbances. Furthermore, the proposed vision-based system combined with the control methodology demonstrates the ability to compensate for mechanical imperfections without changing the morphing hardware. All results collectively illustrate the effectiveness of the proposed IM-NDI in dealing with mechanical imperfections existing in the morphing wing system.

In summary, the overall results confirm two points, namely that (i) adaptability of IM-NDI lead to better robustness against disturbance and backlash, (ii) limited servo bandwidth impairs the ability to mitigate disturbances and track faster-referred signals, and (iii) the effectiveness of compensation can be achieved without the requirement for invasive changes to the design of the morphing mechanism.

REFERENCES

[1] B. Sun, T. Mkhoyan, E.-J. Van Kampen, R. De Breuker, and X. Wang, "Vision-Based Nonlinear Incremental Control for A Morphing Wing with Mechanical Imperfections," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–13, 2022.

[2] T. A. Weisshaar, "Morphing aircraft systems: Historical perspectives and future challenges," *Journal of Aircraft*, vol. 50, pp. 337–353, mar 2013.

[3] R. De Breuker, T. Mkhoyan, N. Nazeer, V. L. Stuber, R. M. Groves, S. van der Zwaag, and J. Sodja, "Overview of the SmartX wing technology integrator," *Submitted to AIAA Journal*, 2021.

[4] T. Mkhoyan, N. R. Thakrar, R. De Breuker, and J. Sodja, "Design of a Smart Morphing Wing Using Integrated and Distributed Trailing Edge Camber Morphing," in *ASME 2020 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, pp. –, American Society of Mechanical Engineers, sep 2020.

[5] X. Wang, T. Mkhoyan, I. Mkhoyan, and R. De Breuker, "Seamless Active Morphing Wing Simultaneous Gust and Maneuver Load Alleviation," *Journal of Guidance, Control, and Dynamics*, pp. 1–14, 2021.

[6] Q. Shen, D. Wang, S. Zhu, and K. Poh, "Finite-time fault-tolerant attitude stabilization for spacecraft with actuator saturation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 2390–2405, 2015.

[7] N. Vafamand, M. H. Asemani, A. Khayatiyan, M. H. Khooban, and T. Dragičević, "TS fuzzy model-based controller design for a class of nonlinear systems including

nonsmooth functions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 233–244, 2020.

[8] Z. Lin, S. Lin, S. Wu, G. Ma, and Z. Liang, "Vibration Control of a Flexible Spacecraft System With Input Backlash," *IEEE Access*, vol. 7, pp. 87017–87026, 2019.

[9] L. Sanches, T. A. Guimarães, and F. D. Marques, "Aeroelastic tailoring of nonlinear typical section using the method of multiple scales to predict post-flutter stable LCOs," *Aerospace Science and Technology*, vol. 90, pp. 157–168, 2019.

[10] R. R. Selmic and F. L. Lewis, "Backlash Compensation in Nonlinear Systems Using Dynamic Inversion by Neural Networks," *Asian Journal of Control*, vol. 2, pp. 76–87, Oct. 2008.

[11] G. Lai, Z. Liu, Y. Zhang, and C. L. Philip Chen, "Adaptive Fuzzy Tracking Control of Nonlinear Systems With Asymmetric Actuator Backlash Based on a New Smooth Inverse," *IEEE Transactions on Cybernetics*, vol. 46, pp. 1250–1262, June 2016.

[12] Y.-C. Lai and T.-Q. Le, "Adaptive Learning-Based Observer With Dynamic Inversion for the Autonomous Flight of an Unmanned Helicopter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 3, pp. 1803–1814, 2021.

[13] B. Sun and E.-J. van Kampen, "Intelligent adaptive optimal control using incremental model-based global dual heuristic programming subject to partial observability," *Applied Soft Computing*, vol. 103, p. 107153, 2021.

[14] E. Tal and S. Karaman, "Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness," *IEEE Transactions on Control Systems Technology*, vol. 29, pp. 1203–1218, may 2021.

[15] Y. Zhou, E.-J. van Kampen, and Q. Chu, "Nonlinear adaptive flight control using incremental approximate dynamic programming and output feedback," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 493–496, 2016.

[16] B. Sun and E.-J. van Kampen, "Incremental model-based heuristic dynamic programming with output feedback applied to aerospace system identification and control," in *2020 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 366–371, IEEE, 2020.

[17] X. Wang, E.-J. Van Kampen, Q. P. Chu, and P. Lu, "Stability Analysis for Incremental Nonlinear Dynamic Inversion Control," Jan. 2018.

[18] L. Ye, Q. Zong, J. L. Crassidis, and B. Tian, "Output-Redefinition-Based Dynamic Inversion Control for a Nonminimum Phase Hypersonic Vehicle," *IEEE Transactions on Industrial Electronics*, vol. 65, pp. 3447–3457, Apr. 2018.

[19] Y. Huang, D. M. Pool, O. Stroosma, and Q. Chu, "Long-stroke hydraulic robot motion control with incremental nonlinear dynamic inversion," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 304–314, 2019.

[20] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive real-time clustering method for dynamic visual tracking of very flexible wings," *Journal of Aerospace Information Systems*, vol. 18, no. 2, pp. 58–79, 2021.

[21] E. Pan, X. Liang, and W. Xu, "Development of Vision Stabilizing System for a Large-Scale Flapping-Wing Robotic Bird," *IEEE Sensors Journal*, vol. 20, pp. 8017–8028, jul 2020.

[22] S. Tijmons, G. C. De Croon, B. D. Remes, C. De Wagter, and M. Mulder, "Obstacle Avoidance Strategy using Onboard Stereo Vision on a Flapping Wing MAV," *IEEE Transactions on Robotics*, vol. 33, pp. 858–874, aug 2017.

[23] P. Serra, R. Cunha, T. Hamel, C. Silvestre, and F. Le Bras, "Nonlinear image-based visual servo controller for the flare maneuver of fixed-wing aircraft using optical flow," *IEEE Transactions on Control Systems Technology*, vol. 23, pp. 570–583, mar 2015.

[24] D. Tang, L. Chen, E. Hu, and Z. F. Tian, "A Novel Actuator Controller: Delivering a Practical Solution to Realization of Active-Truss-Based Morphing Wings," *IEEE Transactions on Industrial Electronics*, vol. 63, pp. 6226–6237, oct 2016.

[25] M. Y. Moemen, H. Elghamrawy, S. N. Givigi, and A. Noureldin, "3-D Reconstruction and Measurement System Based on Multimobile Robot Machine Vision," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021.

[26] Y. Wang, M. Shan, Y. Yue, and D. Wang, "Vision-based flexible leader-follower formation tracking of multiple nonholonomic mobile robots in unknown obstacle environments," *IEEE Transactions on Control Systems Technology*, vol. 28, pp. 1025–1033, may 2020.

[27] B. Sun and E.-J. van Kampen, "Reinforcement-Learning-Based Adaptive Optimal Flight Control with Output Feedback and Input Constraints," *Journal of Guidance, Control, and Dynamics*, vol. 44, pp. 1685–1691, Sept. 2021.

[28] QT Brightek, "QBLP670-IR3 3528 PLCC2 IR LED QT-Brightek PLCC Series 3528 PLCC2 IR LED," 2021.

[29] STMicroelectronics, "N-channel 100V - 0.115 Ω - 10A TO-220 low gate charge STripFET™ II Power MOSFET," 2002.

[30] OptiTrack, "OptiTrack - Primex 41 - In Depth."

[31] R. Acharya, "Multidimensional image analysis and mathematical morphology," in *Sixth Multidimensional Signal Processing Workshop*, (Pacific Grove, CA, USA), p. 203, IEEE, 1989.

[32] B. Fu, F. Han, Y. Wang, Y. Jiao, X. Ding, Q. Tan, L. Chen, M. Wang, and R. Xiong, "High-Precision Multicamera-Assisted Camera-IMU Calibration: Theory and Method," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021.

[33] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2 ed., Mar. 2004.

[34] K. Kanatani, Y. Sugaya, and Y. Kanazawa, "Multiview Triangulation," in *Guide to 3D Vision Computation*, pp. 133–147, Cham: Springer International Publishing, 2016. Series Title: Advances in Computer Vision and Pattern Recognition.

[35] R. Hartley and F. Kahl, "Optimal Algorithms in Multiview Geometry," in *Computer Vision – ACCV 2007*, vol. 4843, pp. 13–34, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. Series Title: Lecture Notes in Computer Science.

[36] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Parallel Real-Time Tracking and 3D Reconstruction with TBB for Intelligent Control and Smart Sensing Framework," in *AIAA Scitech 2020 Forum*, p. 2252, American Institute of Aeronautics and Astronautics (AIAA), jan 2020.

[37] OptiTrack, "Motive API NaturalPoint Product Documentation Ver 2.2."

[38] OptiTrack, "OptiTrack - NatNet SDK - Stream motion tracking data across networks."

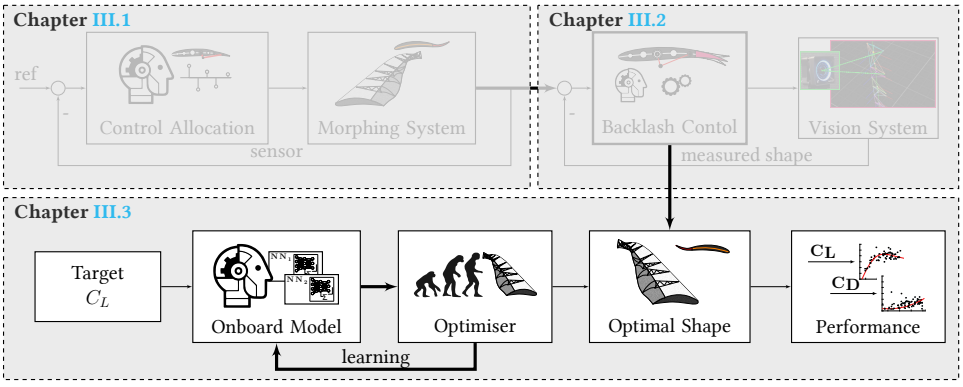
[39] T. Mkhoyan, V. Stuber, X. Wang, I. Mkhoyan, R. De Breuker, and S. van der Zwaag, "Bio-inspired Distributed and Decentralized Real-time Control and Sensing Architecture for a Seamless Active Morphing Wing," *Submitted to Engineering Journal*, 2021.

[40] A. M. Aurand, J. S. Dufour, and W. S. Marras, "Accuracy map of an optical motion capture system with 42 or 21 cameras in a large measurement volume," *Journal of Biomechanics*, vol. 58, pp. 237–240, June 2017.

III.3

Learning-Based Aerodynamic Performance Optimisation – Experimental

In previous Chapters, a control strategy was proposed for load alleviation objectives and mitigation of mechanical imperfections. This Chapter combines these learnings and builds further on Chapter II.2 to address the shape optimisation objective. For this purpose, an online black-box shape optimisation architecture for actively distributed camber morphing wings has been proposed and experimentally validated. Compared to the unmorphed wing baseline shape, a drag reduction of $7.8\,\%$ *was achieved on the SmartX-Alpha demonstrator for a target lift coefficient of* 0.65 *and between* $6.5\,\%$ *and* $19.8\,\%$ *a wide range of other target lift coefficients, with higher drag reductions being associated with lower lift coefficients [1].*



This Chapter is based on the following journal paper:
T. Mkhoyan, O. Ruland, R. De Breuker, and X. Wang, "On-line Black-box Aerodynamic Performance Optimization for a Morphing Wing with Distributed Sensing and Control," *In Review at IEEE Transactions on Control Systems Technology*, 2021

Inspired by nature, smart morphing technologies enable the aircraft of tomorrow to sense their environment and adapt the shape of their wings in flight to minimise fuel consumption and emissions. A primary challenge on the road to this future is how to use the knowledge gathered from sensory data to establish an optimal shape adaptively and continuously in flight. To address this challenge, this Chapter proposes an online black-box aerodynamic performance optimisation architecture for active morphing wings. The proposed method integrates a global online-learned Radial Basis Function Neural Network (RBFNN) model with an evolutionary optimisation strategy, which can find global optima without requiring in-flight local model excitation manoeuvres. The actual wing shape is sensed via a computer vision system, while the optimised wing shape is realised via nonlinear adaptive control. The effectiveness of the optimisation architecture was experimentally validated on an active trailing-edge camber morphing wing demonstrator with distributed sensing and control in an open jet wind tunnel. Compared to the unmorphed shape, a 7.8% drag reduction was realised while achieving the required amount of lift. Further data-driven predictions have indicated that up to 19.8% of drag reduction is achievable and have provided insight into the trends in optimal wing shapes for a wide range of lift targets.

III.3.1. Introduction

Recent trends in aviation highlight the ever-increasing need for fuel economy and sustainability. Active morphing technology can offer significant benefits over conventional wing designs. Due to conflicting requirements [2], conventional wings are only optimised for a single flight condition (such as cruise). By contrast, the ability to morph wings into a desirable shape can allow aircraft to improve flight performance across the full flight envelope actively. While many challenges exist in morphing design, the key challenge to efficiently benefit from active morphing during in-flight operation is how to establish a shape optimisation strategy that is adaptive and can find the global optimum.

The currently practised method of "determining" the optimal wing shape is by selecting from an offline-determined and fixed look-up table. However, the relationship between the wing shape and aerodynamic efficiency depends on many uncertain parameters, which makes the look-up table method suboptimal. By contrast, online data-driven nonlinear optimisation is a promising method, which can change the wing shape adaptively and optimally to any specific flight condition, like birds do [3].

Online shape optimisation strategies for active morphing wings do exist in the literature. In [4], a generative set search method was used to optimise the deflections of eight leading- and trailing-edge control surfaces at a fixed angle of attack (AOA) to reduce the drag on a wind tunnel model. This local black-box optimisation strategy uses a linear lift coefficient model, the parameters of which have to be identified before optimisation through sweeps of the control surfaces for the given angle of attack. The local scope of the linear lift model and the local search character of this black-box direct search method makes this method prone to converge on local optima.

A real-time adaptive least-squares drag minimisation approach has been proposed for the Variable Continuous Camber Trailing Edge Morphing (VCCTEF) concept [5]. It uses a recursive least-squares algorithm to estimate the derivatives of the aerodynamic coefficients with respect to the inputs. The optimal wing shape and elevator deflection are then calculated using the Newton-Raphson method from a constrained optimisation problem. Improvements to the model excitation method, onboard model, and optimisation methods were demonstrated in wind tunnel experiments to achieve up to 9.4 % drag reduction on

the Common Research Model (CRM) with VCCTEF at low subsonic speeds [6]. Simulations have also indicated that a 3.37 % drag reduction is achievable on the CRM with a distributed mini-plain flap system at Mach 0.85 [7].

While the linear-in-the-parameters multivariate polynomial model adopted in [5–7] has a low computational cost, a significant shortcoming is that the model coefficients are only valid around a trimmed equilibrium. The implications of this approach are that the model parameters must be re-identified at every operational point to perform real-time drag minimisation throughout the flight envelope. The model excitation manoeuvres required for re-identification need sweeps of the angle of attack and control surfaces which cause increased fuel consumption and reduced ride comfort. Although the inherent model structure assumptions reduce the model identification cost, they also make the model structure less adaptable. To preserve the same model accuracy, the polynomial model order and the number of coupling terms have to be varied in different flight regimes. Furthermore, gradient-based optimisation methods combined with local models are prone to converge to local optima [8].

The issues of local grey-box model strategies can be overcome by an online black-box global model identification and optimisation approach, which has two potentials: first, global wing shape solutions can be found, leading to more effective drag reductions; second, no additional model excitation manoeuvres or changes to the model structure will be needed for operation at various flight conditions. The global optimisation approach, however, does not come without challenges.

Global optimisation methods generally require more objective function evaluations than local gradient-based methods, making them impractical for direct application to complex aerodynamic shape optimisation. Low evaluation cost global surrogate models may provide a solution by sample-efficiently generalising the information gathered by onboard sensors [9, 10]. Surrogate modelling methods in the literature include Polynomial Regression (PR) [11], Artificial Neural Networks (ANNs) [12], Radial Basis Function (RBF) models [13], and Gaussian Process (GP), also referred to as kriging [14, 15]. In an online data-driven framework, ANNs are promising candidates due to their adaptability and ability to approximate complex nonlinear functions. A type of feedforward ANN is the Radial Basis Function Neural Network (RBFNN), which is featured by its ability to handle noisy, multi-parameter, and scattered data as well as its sensitivity and adaptability to fresh data [16]. These characteristics make RBFNNs promising for in-flight aerodynamic model identification.

Genetic algorithms, also known as evolutionary algorithms, solve optimisation problems by mimicking natural evolution. Analogous to the survival of the fittest principle in evolutionary biology, only the highest quality solutions and their offspring are selected for further consideration. Evolutionary algorithms can deal with discontinuities in the objective function and are suitable for multi-modal and high-dimensional problems [17]. They have been demonstrated to be effective global black-box optimisation tools combined with surrogate models for aerodynamic shape optimisation problems [18, 19]. However, to the best of the authors' knowledge, the existing applications of evolutionary algorithms to the aerospace field are only limited to offline design optimisation problems. Exploiting the merits of evolutionary algorithms in online and real-world scenarios remains an open challenge.

After a global optimal wing shape is determined in flight, a real-time challenge still exists in realising this optimised shape. Intuitively, a feedforward mapping between the servo angle and the trailing-edge displacement can be utilised. However, owing to the nonlinear couplings between aerodynamics and structural dynamics, this mapping is uncertain and is perturbed by external disturbances. Moreover, in an earlier validation experiment of a morphing wing prototype [20], a nonlinear phenomenon named backlash was observed [21, 22]. Owing to the backlash, the output of the morphing mechanism not only depends on the actuator inputs at the current instant but is also determined by the actuation history, leading to an undesirable hysteresis phenomenon [22, 23]. One promising approach to robustly observe and regulate the motions of a physical system is vision-based control, which has shown its effectiveness in aircraft position tracking [24] and inverted pendulum stabilisation [25]. However, the effectiveness of vision-based control in dealing with nonlinear hysteresis and wind disturbances remains unknown.

This Chapter proposes and experimentally validates an effective and adaptable online performance optimisation architecture for active morphing wings. The main contribution is threefold.

1. An online optimisation framework integrating an evolutionary optimisation strategy with a global online-learned radial basis function neural network model for global optimisation, which does not require local model excitation manoeuvres after offline training;
2. Realises the global optimal shape in the presence of backlash hysteresis, model uncertainties, and external gust disturbances using a nonlinear adaptive control algorithm supported by real-time computer vision sensing;
3. Validates the proposed architecture on a seamless active morphing wing demonstrator with distributed sensing and control in an open jet wind tunnel.

This paper is organised as follows. Section III.3.2 formulates the problem. The optimisation architecture is then designed in Sec. III.3.3. The optimised shape realisation using a distributed vision-based control is proposed in Sec. III.3.4. The experimental setup and results are presented in Sec. III.3.5 and Sec. III.3.6, respectively. Finally, conclusions are drawn in Sec. III.3.7.

III.3.2. Preliminaries and Problem Formulation

This section describes the formulation of the online optimisation problem for the morphing wing.

III.3.2.1. Morphing Wing

The active morphing wing considered in this research is illustrated in Fig. III.3.1[1]. The wing has six distributed translation-induced camber modules, allowing the independent camber and spanwise twist morphing [20]. There are twelve servos (two per module) embedded in the wing box, allowing the trailing-edge bottom skin to slide in chord-wise and spanwise directions along a guided sliding interface. Elastomeric skin segments are

[1]The project video can be found via https://www.youtube.com/watch?v=SdagIiYRWyA&t=319s

integrated between the modules to reduce noise and drag. As shown in Fig. III.3.1, the rotational motion of the servo is converted to the sliding motion of the skin by a ball joint linkage system, which results in active morphing of the wing from its nominal NACA6510 shape [20].

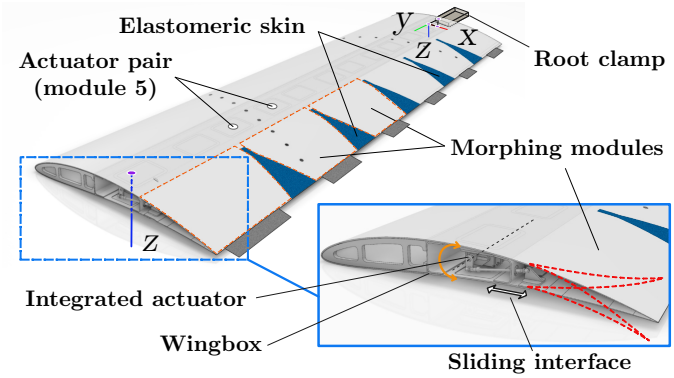


Figure III.3.1: Morphing wing actuation mechanism and coordinate system.

III.3.2.2. Online Optimisation Problem

The goal of online shape optimisation is to find the most aerodynamically efficient wing shape and angle of attack combination without altering the intended flight path of the aircraft. This task is complicated by the uncertainties, unsteadiness, and nonlinearities in aerodynamics. The wing shape is governed by a virtual shape input vector $\mathbf{u} \in \mathbb{R}^{q\times 1}$, which will be elaborated in Sec. III.3.3.1. The wing angle of attack (AOA) is denoted as α. The mappings from $\alpha, \mathbf{u}$ to the lift and drag coefficients C_L and C_D are highly nonlinear and depend on many uncertain parameters. The maximisation of lift-to-drag ratio $\frac{C_L}{C_D}$ results in reduced fuel consumption of the aircraft. Moreover, the right amount of lift force must be generated to maintain level flight, which is determined by the target lift coefficient C_{L_t}. Therefore, the objective of the optimiser is to find the set of inputs $\alpha, \mathbf{u}$ that maximises $\frac{C_L}{C_D}$ while meeting the target lift coefficient C_{L_t} without violating the AOA and actuation limits, which is formulated as

$$\begin{aligned} \underset{\alpha,\mathbf{u}}{\textbf{MINIMISE}} \quad & \frac{C_L(\alpha,\mathbf{u})}{C_D(\alpha,\mathbf{u})} \\ \textbf{SUBJECT TO} \quad & C_L(\alpha,\mathbf{u}) = C_{L_t} \\ & \alpha \in [\alpha_{\text{MIN}}, \alpha_{\text{MAX}}] \\ & \mathbf{u} \in [\mathbf{u}_{\text{MIN}}, \mathbf{u}_{\text{MAX}}] \end{aligned} \tag{III.3.1}$$

This optimisation problem is nonlinear and non-convex because C_L and C_D are nonlinear and non-convex functions of α and $\mathbf{u}$.

It is important to note that although the cost function is formulated as a nonlinear convex cost function, the relationship between the predicted lift and drag coefficients is non-convex and complex. The latter parameters depend on the angle of attack, α, and the

virtual shape inputs, $\mathbf{u}$. Therefore, although the relationship between C_L, C_L and J is convex, the relationship between α, u and C_L, C_L is non-convex. This necessitates the use of nonlinear, non-convex optimisers capable of interacting with global models.

III.3.3. Optimisation Architecture

To solve the online optimisation problem formulated in Sec. III.3.2.2, the input $\mathbf{u}$ is first elaborated in Sec. III.3.3.1 using virtual input functions. This is followed by the cost function design in Sec. III.3.3.2. The optimisation strategy, using the Covariance Matrix Adaptation – Evolutionary Strategy (CMA-ES) and Radial Basis Function Neural Network (RBFNN) onboard model are designed in Sec. III.3.3.3 and Sec. III.3.3.4, respectively. Finally, the overall optimisation architecture is presented in Sec. III.3.3.5.

III.3.3.1. Virtual Shape

The optimisation algorithm is augmented with virtual shape functions to ensure a smooth spanwise representation of the wing shape. The virtual inputs are the parameters of an approximation function that describe the wing shape utilising an input vector $\mathbf{u} \in \mathbb{R}^{q\times 1}$ instead of the actuator inputs. The wing shape is normally represented by the trailing-edge (TE) displacements at the twelve actuator locations $\boldsymbol{z} = [z_1, z_2, ..., z_{12}]^{\mathrm{T}}$. These parameters can be chosen to be less than the number of actuators while still ensuring smooth shape, as was shown in the previous Chapter, Chapter III.1. These virtual shape functions have been used effectively in Chapter III.1 to augment the quadratic solver in the Incremental Nonlinear Dynamic Inversion with the Quadratic Programming (INDI-QP-V) approach for simultaneous gust and manoeuvre load alleviation objective with smoothness constraints.

In the context of the current wing shape optimisation approach, the virtual shape functions have an essential role in decoupling the optimisation problem's dimensionality from the morphing wing's input dimensionality, thereby reducing the computation load substantially.

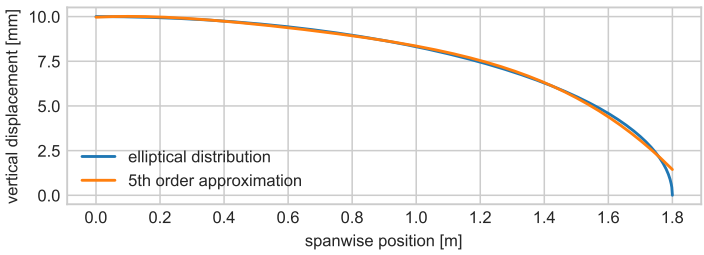


Figure III.3.2: Comparison of an elliptical distribution and 5^{th} order Chebyshev polynomial approximation.

Consistent with Chapter II.2 and III.1, Chebyshev polynomials were chosen as the parametric wing shape approximation function due to their nearly optimal property and orthogonality [26]. The virtual inputs are the parameters of this polynomial function that scale q basis shapes, which are described by the first q Chebyshev polynomials of the first kind: $T_q(x) = \cos(q \cdot \arccos(x))$, where q is a non-negative integer. These polynomials

are orthogonal in the interval $[-1, 1]$ and are re-scaled onto the $[0, 1.80]$ m domain, where 1.8 m is the half-wing span. In other words, the five virtual inputs $u_1, u_2..., u_5$ are the coefficients of a fifth-order Chebyshev approximation of the spanwise camber distribution function that describes the morphed wing shape. The order of the virtual shape function was chosen to be $q = 5$ because it resulted in the greatest reduction in computational loads without compromising the approximation power below acceptable levels for the expected wing shapes. Fig. III.3.2 demonstrates that the 5^{th} order Chebyshev polynomial has sufficient approximation for an elliptical distribution of the 1.8 m wing span.

This distribution function is given as

$$z(y) = \sum_{i=1}^{5} u_i T_i(y) \tag{III.3.2}$$

in which z is the trailing-edge (TE) vertical displacement as a function of the spanwise location y (Fig. III.3.1). The local TE displacement z_i at the i-th actuator is $z_i = z(y_i)$, where y_i is the spanwise location of the actuator. The shapes described by these basis polynomials and their contributions to the amount of camber at the actuator locations are shown in Fig. III.3.3.

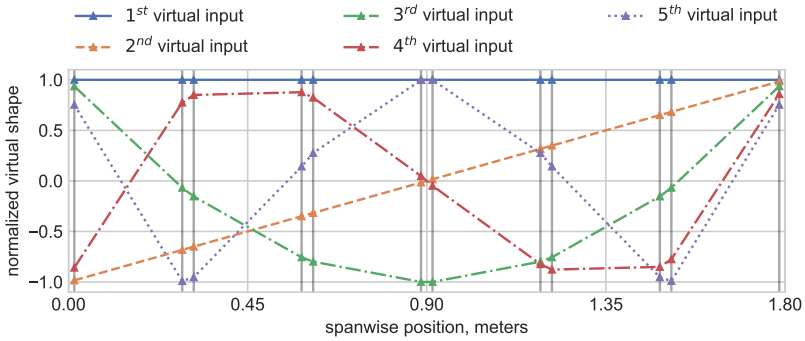


Figure III.3.3: Virtual input basis functions.

Figure II.2.4 in Chapter II.2 shows that most reduction in the Root Mean Square Error (RMSE) is achieved using the first five q polynomials, and the approximation RMSE is approximately 0.1 mm. In contrast, the reduction in RMSE for higher order ($q > 5$) stagnates, but the computational load increases significantly. The following sections explain the choices for the optimisation strategy, using CMA-ES and the onboard model, using RBFNN within the optimisation framework. Concerning the cost of this optimisation framework, the overall benefit of using the virtual shape functions is threefold, namely, (i) a smaller, more efficient onboard RBFNN model can be used, which requires less time to train and update, (ii) the RBFNN model can be evaluated faster, and (iii) the computational cost to populate candidate solutions by CMA-ES is decreased. These aspects collectively reduce the overall optimisation cost of the framework. More results regarding the RMSE on the experimental data are discussed in Sec. III.3.6.3.

III.3.3.2. Cost Function

The cost function is used to score the desirability of the system outputs C_L, C_D. It is designed to maximise the lift-to-drag ratio while regulating the error between the actual and the target lift coefficients, as presented in Eq. (III.3.3).

$$J(C_L, C_D, C_{L_t}) = \underbrace{-\frac{C_L}{C_D}}_{\text{efficiency}} \cdot \underbrace{\frac{k_2}{k_1 + (C_L - C_{L_t})^2}}_{\text{deviation from lift target}} \tag{III.3.3}$$

The efficiency and lift target terms are multiplied such that a low cost can only be reached when high efficiency and the correct amount of lift are achieved simultaneously. A constant $k_1 = 1 \times 10^{-4}$ is added to prevent singularities for small error values. The constant $k_2 = 2 \times 10^{-5}$ serves to scale the cost function output to $[-1, 0]$. As an example, the cost function for $C_{L_t} = 0.50$ is shown in Fig. III.3.4.

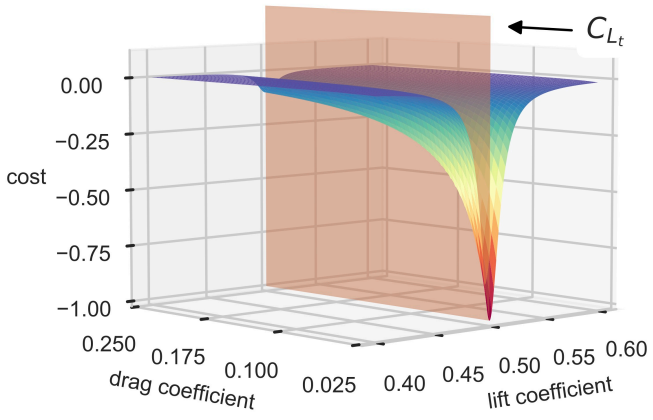


Figure III.3.4: The cost function for $C_{L_t} = 0.50$.

It can be observed from Fig. III.3.4 that a solution that deviates from the target lift coefficient is undesirable even if it provides low drag. In this research, the virtual inputs and angle of attack bounds pose constraints on the inputs and are implemented directly as constraints on the input space. By contrast, the TE displacement bounds and target lift coefficient pose constraints on intermediate and output variables and are implemented indirectly through the cost function scoring. Suppose the local TE displacement at any actuator location is outside of the bounds. In that case, the cost of the associated inputs becomes $J = \max(|z|)^2 + C_J$, in which C_J is a significant positive constant such that the cost will always be higher than those of inputs that do not violate the constraints. The square of the maximum absolute local TE deflection provides a cost gradient to aid the optimiser in steering the solution back to the feasible space.

III.3.3.3. Covariance Matrix Adaptation – Evolutionary Strategy

To solve the nonlinear and non-convex morphing wing shape optimisation problem, while considering online calculation efficiency, the CMA-ES [27] is adopted to perform optimisation. CMA-ES is an evolutionary optimisation strategy for black-box optimisation of nonlinear non-convex continuous problems. With sufficiently large population sizes, CMA-ES has desirable global search performance [27]. CMA-ES iteratively samples populations of candidate solutions from a multivariate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{C})$, which is uniquely identified by its mean $\mathbf{m} \in \mathbb{R}^n$ and covariance matrix $\mathbf{C} \in \mathbb{R}^{n\times n}$ [28]. Based on the returned costs of these candidate solutions, the mean and covariance matrix of the next generation's population are deterministically adapted. This process is repeated until the variation of the cost function converges to below a set threshold.

With g the generational counter, the k^{th} offspring from the subsequent generation $g+1$ is sampled from a multivariate normal distribution $\mathcal{N}$, which is dependent on the current generation's mean search distribution value $\mathbf{m}^{(g)}$, overall standard deviation or step size $\sigma^{(g)}$, and covariance matrix $\mathbf{C}^{(g)}$ as shown in Eq. (III.3.4).

$$\boldsymbol{x}_k^{(g+1)} \sim \mathcal{N}(\mathbf{m}^{(g)}, (\sigma^{(g)})^2, \mathbf{C}^{(g)}), \quad \text{for } k = 1, \ldots, \lambda \tag{III.3.4}$$

The distribution of $\mathcal{N}(\mathbf{m}^{(g)}, (\sigma^{(g)})^2, \mathbf{C}^{(g)})$ is equal to $\mathbf{m}^{(g)} + \sigma^{(g)}\mathcal{N}(\mathbf{0}, \mathbf{C}^{(g)})$. In Eq. (III.3.4), $\mathbf{m}^{(g)}$ shifts the centre of the multivariate normal distribution in a n-dimensional space; $\sigma^{(g)}$ scales the size of the distribution; $\mathbf{C}^{(g)}$ adapts the shape of the distribution. During each iteration of the algorithm, $\mathbf{m}^{(g)}$, $\sigma^{(g)}$, and $\mathbf{C}^{(g)}$ are updated based on the object parameter variations. As shown in Eq. (III.3.5), the mean of the next generation is a weighted average of the μ best scoring search points from the sample $\mathbf{x}_1^{(g+1)}, \ldots, \mathbf{x}_\lambda^{(g+1)}$. In other words, the centre of the next generation's distribution is shifted in the average direction of the best performing candidates:

$$\mathbf{m}^{(g+1)} = \sum_{i=1}^{\mu} w_i \, \mathbf{x}_{i:\lambda}^{(g+1)} \tag{III.3.5}$$

The adaptation law for the covariance matrix is shown as

$$\begin{aligned} \mathbf{C}^{(g+1)} \quad = \quad & (1 - c_{\text{cov}})\mathbf{C}^{(g)} + \frac{c_{\text{cov}}}{\mu_{\text{cov}}} \underbrace{\mathbf{p}_c^{(g+1)} \mathbf{p}_c^{(g+1)^{\mathrm{T}}}}_{\text{rank-one update}} \\ & + c_{\text{cov}} \left(1 - \frac{1}{\mu_{\text{cov}}}\right) \times \underbrace{\sum_{i=1}^{\mu} w_i \left(\frac{\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}\right) \left(\frac{\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}\right)^{\mathrm{T}}}_{\text{rank-}\mu\text{ update}} \end{aligned} \tag{III.3.6}$$

in which c_{cov} and μ_{cov} are the learning rate for updating the covariance matrix and weighting parameter between rank-one and rank-μ updates, respectively. The rank-μ update uses information from previous generations to improve the reliability of the covariance matrix estimator for small population sizes.

The rank-one update exploits the directional information from past generations using the evolution path $\mathbf{p}_c^{(g+1)}$, which is a sum of successive steps defined as

$$\mathbf{p}_c^{(g+1)} = (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{c_c(2 - c_c)\mu_{eff}}\frac{\mathbf{M}^{(g+1)} - \mathbf{M}^{(g)}}{\sigma^{(g)}} \tag{III.3.7}$$

where c_c is the learning rate for cumulation for the rank-one update; μ_{eff} is the variance effective selection mass defined as

$$\mu_{eff} = \left(\sum_{i=1}^{\mu} w_i^2\right)^{-1} \tag{III.3.8}$$

The overall standard deviation $\sigma^{(g)}$ scales the size of the search distribution based on the length of the evolution path compared to its expected length under random selection as

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\left\|\mathbf{p}_\sigma^{(g+1)}\right\|}{\mathrm{E}\,\|\,\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1\right)\right) \tag{III.3.9}$$

in which c_σ and d_σ are the learning rate for the cumulation for the step size control and a damping parameter, respectively. The scaling of the distribution with $\sigma^{(g)}$ can be used to either broaden the distribution's search space or to focus it. When an evolution path is relatively long, the successive steps are roughly in the same direction and the step size should be increased so that fewer iterations are needed to cover the distance. Conversely, when the evolution path is short, the successive steps, at least partially, cancel out each other and the step size should be decreased [28].

III.3.3.4. Onboard Model – Radial Basis Function Artificial Neural Networks

To reduce the online objective function evaluation cost, a global surrogate model (also referred to as the onboard model) is established to approximate the mappings from the angle of attack and wing shape inputs to the lift and drag coefficient outputs using two RBFNNs. These ANNs use RBFs as activation functions and are widely used as function approximators, particularly suitable for establishing multivariate and nonlinear mappings from multi-parameter, noisy, and scattered data sets [16].

The architectures of the two single-hidden-layer RBFNNs used in this research are shown in Fig. III.3.5. Both of them use six inputs: the angle of attack α and the five virtual inputs that describe the wing shape $u_1, \ldots, u_5$. Their respective outputs are the lift and drag coefficients C_L and C_D.

Equation (III.3.10) represents the hidden unit activations given by the basis functions ϕ_j (e.g., Gaussian basis functions). These depend on the input activations from the previous layer $\boldsymbol{x}$, and on the parameters $\boldsymbol{\mu}$ and σ_j [16], where $\boldsymbol{\mu}$ represents the RBF location in $\mathbb{R}^{q+1}$ and σ_j denotes the RBF radius.

$$\phi_j(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{\mu_j}\|^2}{2\sigma_j^2}\right) \tag{III.3.10}$$

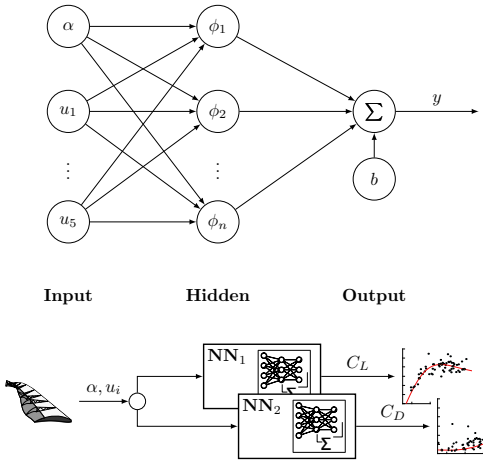


Figure III.3.5: Single-hidden-layer radial basis function neural network architecture.

Both the centre locations and the radii of the RBF basis functions are determined by the network training process. To train the onboard model, the adaptive gradient algorithm (Adagrad) is used [29], which maintains and adapts one learning rate for each dimension using historical data as

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i} \tag{III.3.11}$$

in which $g_{t,i}$ is the gradient of the loss function w.r.t. parameter i at time t, and $G_{t,ii} = \sum_{\tau=1}^{t} g_{\tau,i}^2$ is the cumulative sum of the squares of the past gradients. The effect of $G_{t,ii}$ in Eq. (III.3.11) is that the effective learning rate is diminished over time. Since this diminishing depends on the parameter gradient histories, higher learning rates are used for parameters relating to infrequent features, whereas lower learning rates are used for parameters relating to frequent features. This makes Adagrad well-suited for dealing with sparse data.

The chosen onboard model approach using the trained RBFNN is consistent with the approach used in Chapter II.2. Here it was also shown that the RBFNN onboard model can generalise and adapt online to a secondary wing model, similar in configuration but different aerodynamic baseline shape. This demonstrated the ability to generalise to uncertainties in the wing model, making the structure suitable for global evaluation of the experimental morphing wing model.

III.3.3.5. Optimisation Architecture and Algorithms

An overview of the optimisation architecture is shown in Fig. III.3.6. The left-hand side of Fig. III.3.6 presents a fast model optimisation loop, which comprises three main parts: the cost function (Sec. III.3.3.2), the optimiser (Sec. III.3.3.3), and the onboard model (Sec. III.3.3.4). A pseudocode description of the online optimisation algorithm is given by Algorithm 1.

During each iteration of the optimisation procedure, the evolutionary optimiser generates a population of λ candidate solutions and queries the onboard model (RBFNNs) with

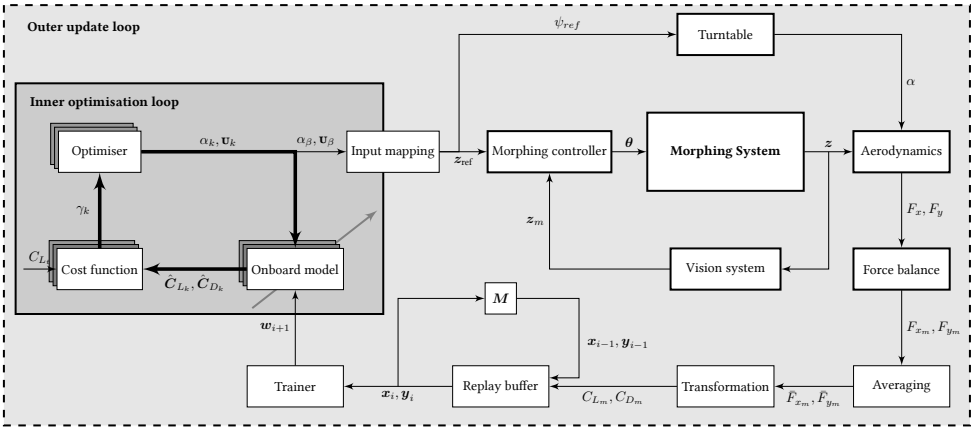


Figure III.3.6: Online shape optimisation architecture diagram with inner optimisation loop and the corresponding components marked in dark gray.

Algorithm 1: Online morphing shape optimisation.

Input: C_{L_t}
Output: C_{L_m}, C_{D_m}
Initilialize: $\epsilon = 1.0$
while RUNNING **do**
  while CALCULATING THE RUNNING AVERAGES $\bar{F}_{x_m}, \bar{F}_{y_m}$ **do**
    while $\epsilon > 1 \times 10^{-6}$ **do**
      Generate a population of λ candidates;
      for $k = 1, \ldots, \lambda$ **do**
        Compute candidate $\hat{C}_{L_k}, \hat{C}_{D_k}$ using RBFNNs;
        Evaluate cost function $\gamma_k = J(\hat{C}_{L_k}, \hat{C}_{D_k}, C_{L_t})$;
      end
      $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_\lambda]$;
      $\epsilon = \text{MAX}(\boldsymbol{\gamma}) - \text{MIN}(\boldsymbol{\gamma})$;
    end
    Define the most promising candidate as the β-th that leads to $\text{MIN}(\boldsymbol{\gamma})$;
    Transform $\mathbf{u}_\beta \rightarrow \boldsymbol{z}_{ref}$;
    $\psi_{ref} = \alpha_\beta - \varepsilon$;
    Command ψ_{ref} and $\boldsymbol{z}_{ref}$;
  end
  Transform $\bar{F}_{x_m}, \bar{F}_{y_m}$ to C_{L_m}, C_{D_m};
  Add C_{L_m}, C_{D_m} to training buffer;
  Retrain RBFNNs;
end

their angle of attack and wing shape combinations $\alpha_k, \mathbf{u}_k$, where $k = 1, \ldots, \lambda$. In turn, the onboard model predicts the steady-state lift and drag coefficients $\hat{C}_{L_k}, \hat{C}_{D_k}$ resulting from each of these inputs, and the cost function scores the desirability $\gamma_k = J(\hat{C}_{L_k}, \hat{C}_{D_k}, C_{L_t})$ of these predicted outputs. Subsequently, the scores of the evaluated input combinations are used by the optimiser to generate a more promising population of input combinations. This cycle is repeated with a frequency of approximately 15 Hz, depending on the population size and available computing power. This process continues until the optimiser converges onto a single most optimal $\alpha, \mathbf{u}$ combination, whose index is defined as β. This

input combination is then actuated on the real system and the resulting measurements are subsequently used to improve the onboard model.

Since the zero position of the turntable did not coincide with the zero AOA, the turntable angle to be commanded to realise any true aerodynamic angle of attack was unknown beforehand. To address this challenge, the angular difference ε between the commanded turntable angle ψ_{ref} and the aerodynamic angle of attack α, was estimated using measurements of a sweep in AOA with the wing jig (baseline) shape and the iterative approach outlined in Algorithm 2. During this procedure, the turntable misalignment constant ε was iteratively estimated by matching the measurement-based estimated zero-lift AOA $\hat{\alpha}_{C_L=0}$ to the theoretical zero-lift AOA $\alpha^*_{C_L=0}$. This theoretical zero-lift AOA was determined to be -6.5570 degrees using the 3D panel viscous solver *XFLR5 v 6.48*. This estimation procedure was performed once at the beginning of the experiment.

The optimisation problem is required to optimise the angle of attack and virtual inputs for a given target lift coefficient while maximising the steady-state lift-to-drag ratio. Therefore, a limited but considerable time window could be reserved to find the solution iteratively. The selection of the population size was made in the effort to utilise the given time window for calculation fully. With the selected system and population size, convergence was achieved on average within 10.7s and 7.5s on Intel Core i7-4510U Central Processing Unit (CPU), 8.00 GB Random-access Memory (RAM) and Intel Xeon W-2223 CPU 3.60 GHz, 16.00 GB RAM, respectively. The optimiser's convergence robustness was evaluated in Chapter II.2, Sec. II.2.3.2 (Fig. II.2.13) and discussed further in Sec. III.3.5.3 for the experimental system and chosen optimiser configuration.

Algorithm 2: Turntable misalignment estimation.

Input: $\psi, F_{x_m}, F_{y_m}, \alpha^*_{C_L=0}$
Output: ε
Initilialize: $\epsilon = 1.0,\ \varepsilon = 1.0$
while $\epsilon > 1 \times 10^{-8}$ **do**
  Adjust ε based on error ϵ: $\varepsilon = \varepsilon + 0.5 \cdot \epsilon$;
  Calculate angles of attack $\alpha = \psi + \varepsilon$;
  Transform F_{x_m}, F_{y_m} to C_L, C_D using α;
  Estimate $\partial C_L / \partial \alpha$;
  Estimate zero lift angle of attack $\hat{\alpha}_{C_L=0}$;
  Update error: $\epsilon = \hat{\alpha}_{C_L=0} - \alpha^*_{C_L=0}$;
end

Using the ε estimation, the most promising aerodynamic AOA α_β is converted to an equivalent turntable angle reference $\psi_{ref} = \alpha_\beta - \varepsilon$. In addition, the virtual shape inputs $\mathbf{u}_\beta$ are transformed to the reference TE displacements at the twelve actuator locations $\boldsymbol{z}_{\text{ref}} \in \mathbb{R}^{12}$. Then, the reference table angle and trailing edge displacements are sent to the turntable and morphing controllers, respectively. The morphing controller uses the TE displacement feedback $\boldsymbol{z}_m$ from a vision system to steer the TE displacements to their reference values by controlling the morphing actuators $\boldsymbol{\theta}$. The vision system and the morphing controller will be elaborated in Sec. III.3.4.

After the controllers have converged and the intended wing shape and angle of attack are actuated on the system, the resulting aerodynamic forces F_{x_m}, F_{y_m} are measured using a force balance which is mounted to the turntable. The 40-second running averages of the

resulting measurements denoted as $\bar{F}_{x_m}, \bar{F}_{y_m}$ are then transformed to the lift and drag coefficients C_{L_m}, C_{D_m} using an α-dependent nonlinear mapping. Both the inputs used and the resulting aerodynamic coefficients are added to the training set, which is kept in memory. Finally, the loop is closed by training the onboard model, which results in updated parameters $\boldsymbol{w}_{i+1}$.

III.3.3.6. Summary and remarks

It is important to note that although the cost function is formulated as a nonlinear convex cost function (see Fig, III.3.4 for $C_{L_t} = 0.50$), the relationship of the predicted lift and drag coefficients is non-convex, and depend on the angle of attack, α, and the virtual shape inputs, $\mathbf{u}$. For this reason, although the relationship between C_L, C_D and J is convex, the relationship between α, u and C_L, C_D is non-convex. This necessitates the use of nonlinear, non-convex optimisers capable of optimising global models.

Furthermore, given that the aerodynamic model on which the onboard model is trained is data-driven, candidate shape and angle of attack solutions cannot be formulated explicitly for the given lift coefficient. Therefore, the candidate solutions are proposed to be evaluated iteratively in the inner optimisation loop. This brings the importance of the second part, namely the need for a low-cost data-driven onboard model which can be updated globally in an outer loop using the measurements from the actual shapes evaluated online. For this purpose, the low-cost RBFNN is proposed with local sensitivity to measurement updates and the capability to be trained on the actual morphing wing model in a real-world experiment.

In summary, the optimisation architecture shown in Fig. III.3.6 two essential components, namely, the fast inner optimisation loop and the slower outer loop, which are responsible for updating the global RBFNN model based on new measurements of the lift and drag coefficient obtained from the optimised shapes on the experimental wing model. The fast inner loop efficiently evaluates the candidate solutions of virtual shapes and angles of attack for a given target lift coefficient. Given the current optimisation structure with the fast inner optimisation loop and the synchronisation architecture discussed in Sec. III.3.5.2, the CMA-ES lends itself suitable for implementing a parallel query approach. In this approach, the computational time is further reduced by implementing a query of multiple candidate solutions on the low-cost RBFNN model. This completes the architecture.

III.3.4. Optimised Shape Realisation – A Vision-Based Control Approach

A vision-based control approach is adopted to realise the optimised shape on the real physical system while resisting external disturbances, model uncertainties, and mechanical imperfections. The distributed vision system is described in Sec. III.3.4.1, while an adaptive nonlinear shape control algorithm is presented in Sec. III.3.4.2.

III.3.4.1. Vision System

Vision-based Shape Reconstruction

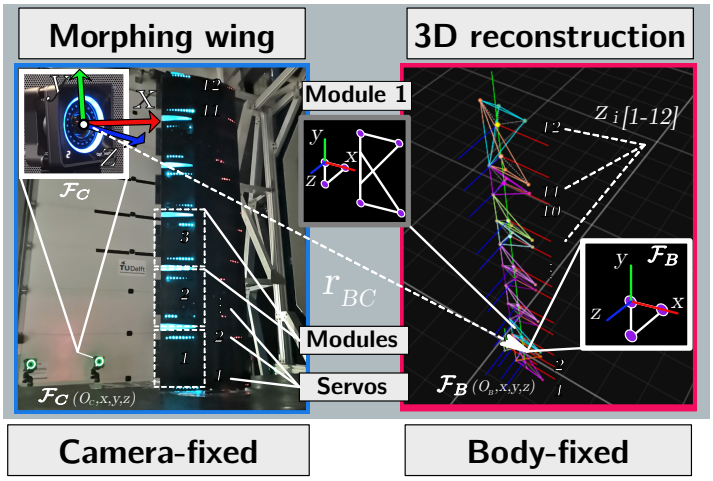


Figure III.3.7: Coordinate systems and vision-based wing shape reconstruction.

A distributed vision system is developed to provide the knowledge of the morphing wing' shape to the controller in real-time. The variable of interest to the controller is the local vertical displacement of the wing trailing edge with respect to a body-fixed coordinate system. The body-fixed coordinate system $\mathcal{F}_B$ is chosen to be near the root of the wing with an origin O_B (Fig. III.3.7). The trailing edge displacement is reconstructed in the $\mathcal{F}_B$ frame in real-time utilising a 5-view vision-based tracking system. Each morphing module is fitted with a pair of active infrared (IR) light-emitting diodes (LEDs), with 3 additional markers for the definition of $\mathcal{F}_B$. An overview of the vision-based tracking pipeline is shown in Fig. III.3.8.

3D Reconstruction

The n-view 3D reconstruction problem is concerned with finding the optimal estimation of an object $\hat{X}$ in a 3D global coordinate frame (i.e., locations in the x, y and z-axes), which is observable in noisy $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$ points correspondence in n camera views. The point correspondences $\bar{x}_i$ are generally defined by markers in u, v coordinates of a 2D image plane and transformed to a camera fixed reference frame $\mathcal{F}_C$ via triangulation [30].

Several successive image processing steps are implemented to refine the observation of point correspondences before this process. The most common setup for triangulation is a calibrated 2-view stereo camera setup, which was demonstrated in a previous study for reconstructing flexible wing motion [31]. However, the tracking accuracy and redundancy can be improved by adding more distributed camera observations. As highlighted in [31], this is particularly beneficial for objects subject to adverse environmental conditions (such as disturbing flow conditions and mechanical vibrations), where calibration drift can be accumulated over time. Therefore, a five-camera setup was adopted in this research.

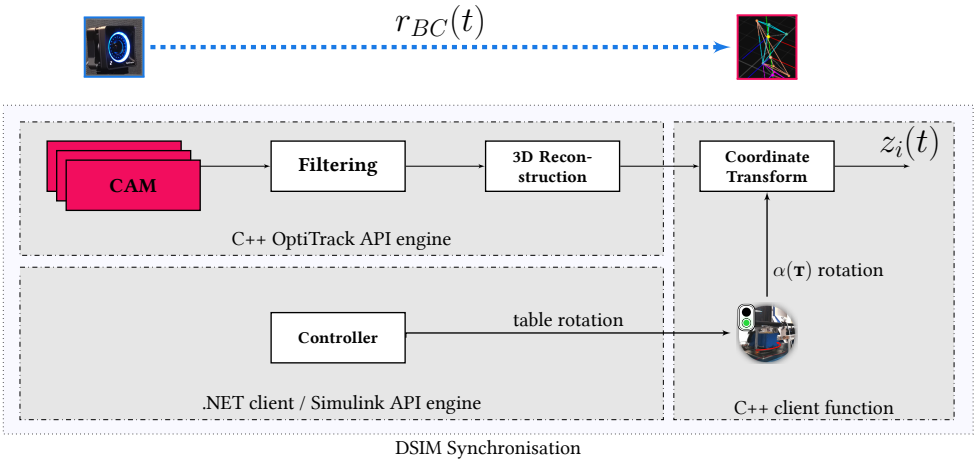


Figure III.3.8: Real-time vision-based tracking pipeline.

The principle of n-view reconstruction relies on back-projecting the 3D point onto the respective camera views, allowing to define a minimisation problem for the re-projection error, $E = \sum_{k=1}^{n} ||x_k - \bar{x}_k||^2$. The n-view minimisation problem is commonly solved by an expanded linear system of equations similar to singular value decomposition (SVD) in a direct linear transform (DLT) procedure [32]. Global optimisation methods can be applied, such as algebraic, matrix inequality, and the $\mathcal{L}_\infty$ approach [33]. More computationally intensive methods also exist, such as bundle adjustment, where the camera calibration parameters are not known a priori and are included in the minimisation problem [30, 32]. This research adopted a proprietary 3D point cloud reconstruction engine in the OptiTrack API system [34]. Multi-camera calibration was performed by a wanding process, resulting in an average calibration error of 0.25 mm for all cameras.

Coordinate Transformation

As illustrated in the 3D view of Fig. III.3.9, a coordinate system transformation from $\mathcal{F}_C$ to $\mathcal{F}_B$ is required for shape reconstruction. This transformation is performed by a translation, followed by 3-axis rotations in pitch, roll, and yaw axes (θ, ϕ, ψ). It is noteworthy that $\mathcal{F}_B$ is attached to the wing and thus rotates along with the turntable. Therefore, the transformations have to be performed continuously in real-time. The applications for processing, reconstructing, and accessing the data are written in a low-level C++ programming language for performance enhancement. The average total processing latency of the complete processing pipeline (Fig. III.3.8) was in the range of 5-7 ms, which is smaller than the control sampling interval (16.67 ms).

III.3.4.2. Nonlinear Adaptive Vision-Based Control

The objective of the morphing controller is to steer the distributed actuators to morph the wing to the optimal shape commanded by the optimiser, using the real shape reconstructed by the vision system as a feedback signal (Fig. III.3.6).

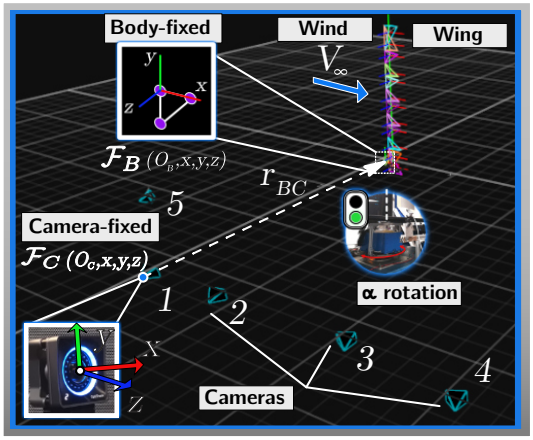


Figure III.3.9: A 3D orientation of the cameras with respect to the morphing wing positioned in the wind tunnel.

The morphing wing can be represented by the following nonlinear multi-input multi-output system:

$$\dot{x}(t) = f(x(t), u(t)) + d(t),\ y(t) = h(x(t)) \tag{III.3.12}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $y \in \mathbb{R}^p$ are the system state, input, and measurable output, respectively. $d(t) \in \mathbb{R}^n$ is the external disturbance vector. For this morphing control problem, the output signal y is provided by computer vision. $f : (\mathbb{R}^n, \mathbb{R}^m) \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are assumed to be Lipschitz continuous on their domains.

To make the controller constantly adapt to changes in the system and environment, online system identification is necessary. A precise online identification method for dynamic systems that only requires output feedback and less computation power than neural networks has been proposed in [35]. This method identifies the so-called incremental [36] model using sampling interval-based linearisation and discretisation.

Theorem 1. *[35] If the nonlinear system given by Eq. (III.3.12) is observable, then under sufficiently high sampling frequency, the output tracking error increment* $\Delta e_{t+1} = \Delta y_{t+1} - \Delta y_{ref,t+1}$ *can be determined uniquely from the observations and control inputs over a sufficiently long time horizon,* $[t - M + 1, t],\ M \geq (n + p)/p$*:*

$$\Delta e_{t+1} \approx F_t \overline{\Delta e}_{t,M} + G_t \overline{\Delta u}_{t,M}, \tag{III.3.13}$$

where $F_t \in \mathbb{R}^{p \times Mp}$ is the augmented transition matrix, and $G_t \in \mathbb{R}^{p \times Mm}$ is the augmented input distribution matrix. Also, $\overline{\Delta u}_{t,M} = [\Delta u_t^{\mathrm{T}}, \Delta u_{t-1}^{\mathrm{T}}, ..., \Delta u_{t-N+1}^{\mathrm{T}}] \in \mathbb{R}^{Nm}$ and $\overline{\Delta e}_{t,M} = [\Delta e_t^{\mathrm{T}}, \Delta e_{t-1}^{\mathrm{T}}, ..., \Delta e_{t-N+1}^{\mathrm{T}}] \in \mathbb{R}^{Np}$ are the input and tracking error data of N previous samples, respectively. Using Eq. (III.3.13), the nonlinear dynamic inversion control in its discrete form can be applied, which regulates the tracking error to zero asymptotically in spite of model uncertainties, external disturbances, and backlash.

III.3.5. Experimental Design and Setup

To assess the performance of the proposed online shape optimisation architecture, an experiment was conducted with the SmartX-Alpha morphing wing at the Open Jet Facility (OJF) wind tunnel [37].

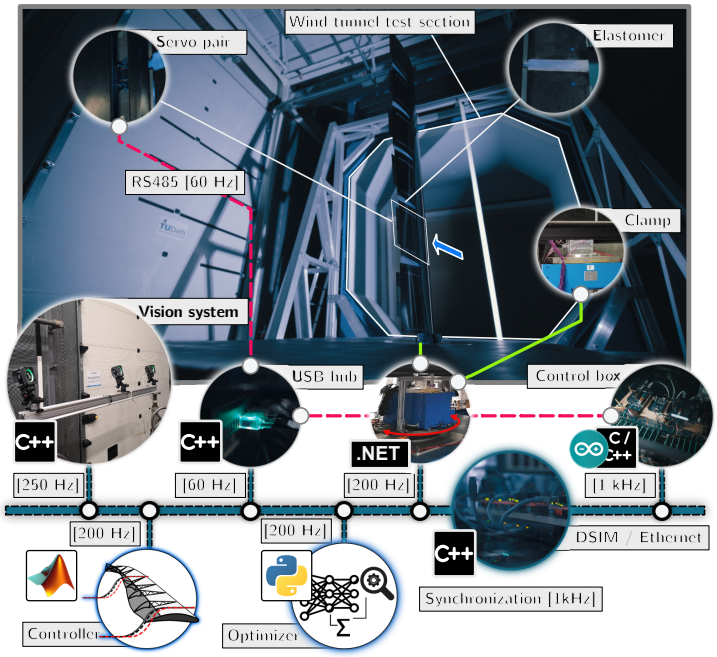


Figure III.3.10: Experimental apparatus with various hardware, software, and vision-based control components. The green, red, and blue lines represent mechanical, electrical, and synchronisation paths, respectively.

III.3.5.1. Apparatus

The experimental apparatus is shown in Fig. III.3.10. The system consists of the SmartX-Alpha morphing wing, mounted vertically on an actively controlled turntable system, and placed in the wind tunnel test section. The operational point was selected as $V = 15$ m/s. The wing is clamped on a three-axis external balance measurements system, allowing to measure root reaction forces and moments at 1000 Hz. The shape command and the resulting lift distribution are achieved by twelve high-performance servos connected to an array of RS-485 devices communicating serially via the RS-485 protocol. The update rate is constrained by the physical USB host interface with a fixed time delay of 15 ms. The actuation angles of the servo are constrained to $\pm 25^\circ$ as not to exceed the physical limits of the morphing system. The online optimisation was carried out on Intel Xeon W-2223 CPU 3.60 GHz Central Processing Unit (CPU) system, 16.00 GB Random access memory (RAM).

An array of IR-LEDs of type 3528 850NM WLP PLCC2, characterised by 850 nm wavelength, is installed on the wing bottom surface and powered by a 12V direct current

(DC) power supply. The brightness is actively controlled by the IRF520 Power metal–oxide–semiconductor field-effect transistor (MOSFET) dimmer circuit. Five Prime^{x}41 4.1 megapixel IR cameras are responsible for marker tracking at a frame rate of 250 frames per second (FPS) [34]. The shape-reconstruction algorithm is written in C++ and deployed on Dell Optiplex 7400 i5-8500 3.0 GHz CPU system, with 8.0 gigabytes (RAM).

To continuously control the wing AOA, a real-time turntable control loop has been implemented. The Franke turntable of type LTB 400 is equipped with a brushless TC-60-1.3 motor with encoder and braking system. An MSR-40-MOR rotary encoder measures the table angle. The turntable is controlled by a proportional–integral–derivative algorithm whose parameters are tuned to provide smooth table angle command tracking while satisfying servo rate and position limits. The servo commands are communicated via RS232 protocol over a USB controller. A .NET-based software control interface is developed to set control parameters and receive the encoder feedback signal, which is interfaced to the synchronisation framework in real-time at 200 Hz. The control interface acted only as the connecting interface between the servo control hardware and the data synchronisation framework. The actual servo control was achieved by the embedded servo controller of the LTB 400 turntable.

III.3.5.2. Real-Time Synchronisation

Various hardware and software components must cooperate coherently and harmoniously in real-time. To facilitate this, a distributed data-sharing architecture was developed based on the decentralised communication principle, which allows parallel integration of hardware and software components in various programming languages (Python, Matlab, Simulink, C++, .NET, etc.) and various communication protocols (RS485, Ethernet, Modbus, etc.). The architecture software is developed in C++ with the real-time D-SIM framework, connecting several PC nodes over a local Ethernet network [38], and enabling synchronisation as depicted in the bottom part of Fig. III.3.10. This approach provides several key benefits over conventional centralised systems: 1) running hardware and software processes in parallel at non-uniform sampling rates with a 1 kHz synchronisation of shared variables between processes; 2) scalability and easy-to-be modified system structure; 3) mixing various programming languages and protocols for various experimental components.

III.3.5.3. Optimisation Configuration

The CMA-ES algorithm was used to solve the optimisation problem. A relatively large population size of $\lambda = 150$ was used to improve the global search performance of the algorithm. The middle of the input domain was used as the initial solution point $\mathbf{x}_0$. To allow global convergence, the initial standard deviation $\boldsymbol{\sigma}_0$ and the scaling of the input variables were selected such that $\mathbf{x}_0 \pm 2\boldsymbol{\sigma}_0$ spanned the width of the domain in each of the input axes. A suitable cost function variation convergence threshold was found at 1×10^{-6}, such that the optimiser yielded adequate convergence considering the computational load. The computational time was further reduced by a parallel query of 150 candidate solutions, improving the approach presented in Chapter II.2. With the selected system and population size, convergence was achieved on average 7.5s on Intel Core i7-4510UCPU, 8.00 GB RAM and Intel Xeon W-2223 CPU 3.60 GHz, 16.00 GB RAM, which was used for the experiment.

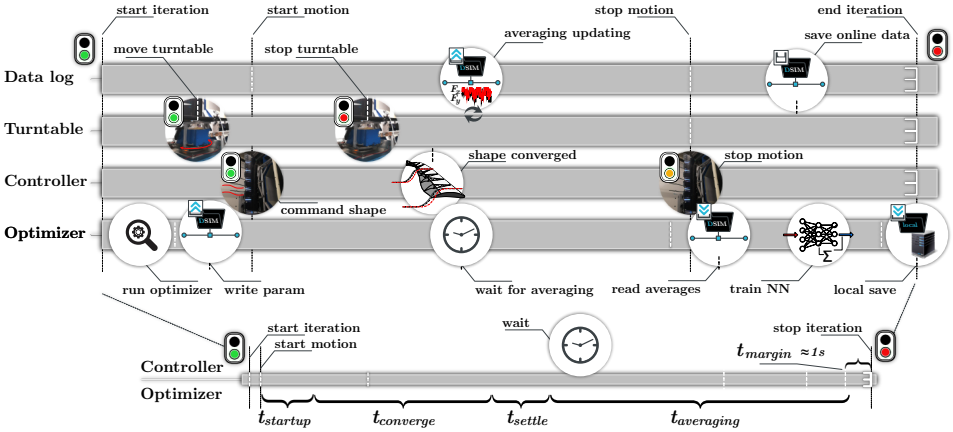


Figure III.3.11: Control flow and timeline of the experimental system.

The onboard model on which the candidate solutions from the optimiser are evaluated comprises two single-hidden-layer RBFNNs. These networks are continuously improved, with training being performed each iteration using training data kept in memory in the replay buffer. After balancing approximation power and computational load, the lift and drag coefficient networks (Fig. III.3.5) was configured with 500 and 940 neurons, respectively. The higher neuron count in the drag coefficient network is necessary for capturing the higher degree of nonlinearities and spanwise distribution dependencies in drag. During training, the RBF centre locations, radii, network weights, and the bias parameter are updated using the Adagrad algorithm with a mean squared error loss function. The initial learning rates for both networks were configured as 0.01.

The training data, comprised of previously evaluated inputs and their lift and drag coefficients, is stored in a buffer. In this research, replacing old data points when the buffer is full is based on the nearest neighbour search on all points in the buffer, inspired by the coverage maximisation strategy presented in [39]. The data point with the lowest mean euclidean distance to its ten closest neighbours is replaced with the latest available data point. This replacement strategy aims to maximise the coverage domain of the training set by replacing the data points in regions of high data density and holding onto samples in data scare regions of the domain.

Because of the backlash effects in the actuation mechanism, the required actuator angles for any given wing shape are not unique and are unknown beforehand. Therefore, limits were imposed on the commanded local z-displacements of the trailing edge at the actuator positions $z_1, z_2, \ldots, z_{12}$. The maximum absolute displacement achievable at any actuator position is dependent on the actuation of the neighbouring actuators. Actuator pairs that deflect in unison can effectuate larger trailing-edge displacements than actuator pairs that deflect in opposite directions. The local vertical displacements allowed for the optimiser were selected as ± 10 mm.

III.3.5.4. Experimental Procedures

The experimental control flow is shown in Fig. III.3.11, responsible for the operation of various system components during the experiment. The order and measurement conditions of the performed runs are shown in Table III.3.1. The three types of runs performed are baseline, wandering, and optimisation. The baseline runs are AOA sweep with a fixed (jig) wing shape to establish a performance baseline. During the wandering phase runs, pseudo-random (PR) inputs were actuated on the system to explore the input space for onboard model identification. Throughout the optimisation runs, the optimal angles of attack and wing shapes established by the optimiser were commanded to maximise the lift-to-drag ratio.

Table III.3.1: Experiment test matrix, measurement with the angle of attack bias in grey (PR = Pseudo-Random).

C_{L_t}	α	wing shapes	type	no. of samples
-	sweep	jig shape	baseline	18
-	PR	PR	wandering	150
0.65	optimal	optimal	optimisation	15
-	sweep	jig shape	baseline	18
-	PR	PR, reduced bounds	wandering	57
0.40	optimal	optimal	optimisation	30
0.75	optimal	optimal	optimisation	30
0.90	optimal	optimal	optimisation	40

III.3.6. Experimental Results and Discussion

In this section, the experimental results are presented and discussed. First, a performance baseline is established, and the wandering phase measurements are presented in Sec. III.3.6.1. Then, in Sec. III.3.6.2, the results from online optimisation for $C_{L_t} = 0.65$ are elaborated upon. Finally, experimental data-driven optimisation predictions are shown in Sec. III.3.6.3 for a wider range of target lift coefficients.

III.3.6.1. Baseline and Wandering Phase

A performance baseline was established by measuring the aerodynamic forces of the wing jig shape at various angles of attack. This wing jig shape was realised by performing a doublet manoeuvre without wind. Subsequently, forty-second averaged force measurements were taken at table angles from -18 to 10 degrees in increments of two degrees at a wind speed of 15 m/s.

The performance baseline was established by interpolating the jig shape measurements with a 16^{th} degree polynomial. This was done to provide a sufficient approximation of the region between 0.4 and 1.2 target lift coefficient. Since additional data was used to construct the polynomial outside of the shown interval in Fig. III.3.12, no oscillations (Runge's phenomenon) were encountered in the evaluated target lift coefficient interval [0.2-1.2].

Both the jig shape measurements and the fitted model are shown in Fig. III.3.12. During the wandering phase, the actuation space was explored with pseudo-random inputs. As shown in Fig. III.3.12, naturally, the performances of the pseudo-random wandering inputs

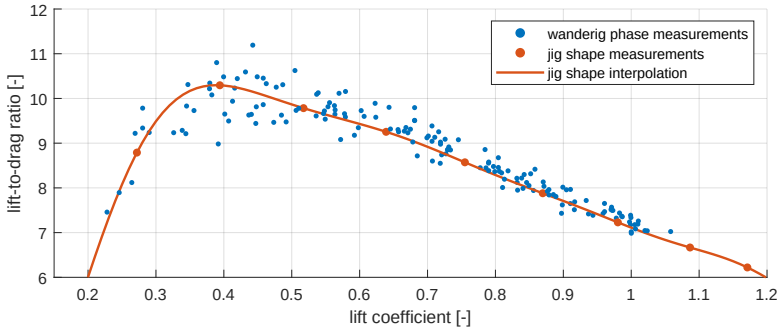


Figure III.3.12: Wing jig shape baseline performance measurements (orange dots), wing jig shape performance fitted model (orange line), and performance measurements from pseudo-random exploration (blue dots).

are distributed around the jig shape performance curve. While the jig shape curve serves as a baseline, the distribution of the wandering phase performances roughly indicates the physical bounds of the attainable performance with active morphing for SmartX-Alpha.

III.3.6.2. Online Optimisation

After the first 150 wandering phase measurements, 15 iterations of online wing shape optimisation were performed with a target lift coefficient $C_{L_t} = 0.65$. The resulting lift and drag coefficient measurements are shown in Fig. III.3.13. During the first six iterations, the discrepancies between the targeted and the measured lift coefficients are relatively large, i.e., ± 0.25. At the same time, the measured drag coefficients also fluctuate considerably. This is to be expected, as the lift-induced drag dominates the total drag. Furthermore, these initially evaluated wing shapes are suboptimal, attributed to the observation of increased amounts of camber near the root and tip. While the true optimal shape is unknown, the amounts of camber at those locations are expected to decrease for constructing a nearly elliptical optimal lift distribution.

After iteration six, the measured lift coefficient converges to its target. Moreover, Fig. III.3.13 shows that the wing shape and AOA combinations evaluated from iteration seven onward not only realised C_{L_t} but also achieved a lower C_D. In other words, the executed wing shapes are more efficient, as indicated by the positive lift-to-drag ratio increase percentage. The most desirable performance was measured for the input combination evaluated during iteration 12 with $C_L = 0.642$. The measured lift-to-drag ratio at this iteration was 10.015, corresponding to a 7.8 % drag reduction compared to the wing jig shape at the same C_L.

The wing shape evaluated at iteration 12 is shown in Fig III.3.14. This shape comprises maximum positive camber between 1.4 and 1.6 m from the wing root (corresponds to module 5), a steep decrease at the wing tip, and a gradual reduction of the local camber towards the root end of the wing. It is mainly the reduction of the local camber near the wing ends, which is supposed to reduce the strength of the wing tip vortices, that results in the observed improvement in the aerodynamic efficiency. These vortices result from the spanwise flow components caused by the ”leaking” of high-pressure air from the bottom side of the wing around the wing tips towards the lower pressure regions on

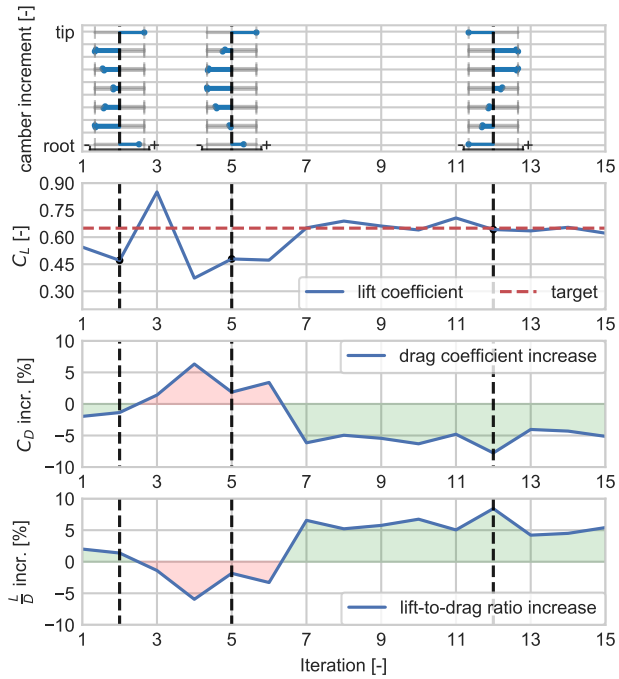


Figure III.3.13: Measured lift and drag coefficients, and selected wing shapes from online optimisation with a target lift coefficient of 0.65 for 15 iterations.

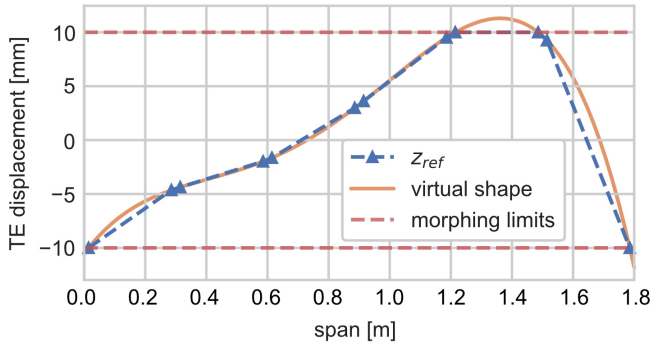


Figure III.3.14: Shape evaluated at iteration 12 during online optimisation with a target lift coefficient of 0.65.

the upper wing surface. Nevertheless, the optimal location of the maximum camber is expected to be between the wing root and the centre of the wing, which corresponds to an interpolation between the elliptical distributions associated with wings with one and two free ends. There, it is suspected that even though the given shape already offers a 7.8 % drag reduction over the wing jig shape for this target lift coefficient, even more efficient wing shapes exist.

An advanced, efficient wing shape is predicted using the measurements from the second wandering phase. After the first optimisation run, the zero angles of the turntable was reset by reconfiguring the turntable hardware and software as part of the shut-down and start-up procedure. The shift in the turntable zero position was approximated by matching the force balance readings to earlier established benchmarks through variation of the table angle. However, later analysis revealed this approximation method to be less accurate than was expected. Post-processing of the measurements using Algorithm 2 showed that the shift in the table angle zero position was overestimated by approximately 1.6 degrees. As a result, the accurate AOA and the AOA-dependent lift and drag coefficients of the second wandering phase were unintentionally biased, affecting the performance of the later optimisation runs negatively. Nevertheless, more accurate optimal wing shape predictions are achieved by correcting these biases in post-processing. These results are presented in Sec. III.3.6.3.

III.3.6.3. Experimental Data-Driven Optimisation Predictions

To make an improved estimation of the optimal shape and its corresponding drag reduction and to make a more general prediction about the optimal wing shapes at other target lift coefficients, the online training was simulated using experimental data collected during wandering phases. In post-processing, the samples measured after the AOA bias was introduced in the turntable were corrected. These samples were then fed to the optimisation algorithm on a per-sample basis to simulate the wandering phase experiment. A validation subset of samples was used to estimate the predictive accuracy of the trained model.

When $C_{L_t} = 0.65$, the optimal wing shape as computed by the optimiser on the trained onboard model, was shown in Fig. III.3.15. With an AOA of 0.8 degrees, the predicted lift-to-drag ratio of this shape is 10.35. This corresponds to an 11.1 % drag reduction compared to the wing jig shape. Furthermore, the shape shown closely represents the expected optimal shape described in Sec. III.3.6.2, with a gradual reduction of airfoil camber towards the free wing tip end and a more moderate amount of camber reduction at the root end as a result of pressure leakage at the root intersection.

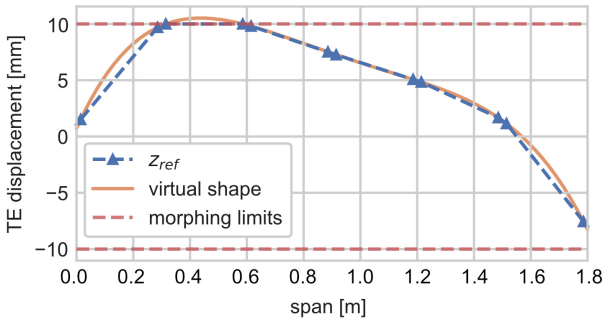


Figure III.3.15: Experimental data-driven optimal shape prediction for $C_{L_t} = 0.65$.

The predictive accuracy of the trained onboard model was evaluated using error measures computed on the validation data set. The RMSE and their normalised counterparts are shown in Table III.3.2. The normalisations were performed using the domain width of the corresponding output variables, i.e., NRMSE $=$ RMSE$/(y_{\text{MAX}} - y_{\text{MIN}})$. Table III.3.2 shows that the NRMSE values of the lift and drag coefficient RBFNN models are close to each other. By contrast, the lift-to-drag ratio NRMSE is higher because the lift-to-drag ratio is not approximated with a dedicated neural network but by the ratio of two estimated outputs. Nevertheless, based on the relatively small training data set, the onboard model can predict the lift and drag coefficients of the validation data set samples with an average prediction error of approximately 2.5 % of their respective domain widths.

Table III.3.2: Model error measures on the validation data set.

	RMSE [-]	NRMSE [%]
C_L	0.0147	2.34
C_D	0.0026	2.50
L/D	0.2818	7.75

To reveal the trend in the estimated optimal wing shape, shape optimisation was also conducted for target lift coefficients of 0.35, 0.50, and 0.80. The predicted optimal shapes for these lift coefficients are shown in Fig. III.3.16. Generally, the maximum amount of camber morphing is commanded at approximately one-quarter span, with a gradual reduction approaching the minimum camber limit towards the wing tip. For higher target lift coefficients, the area under the virtual shape curves, which can be considered the overall amount of camber, is increased.

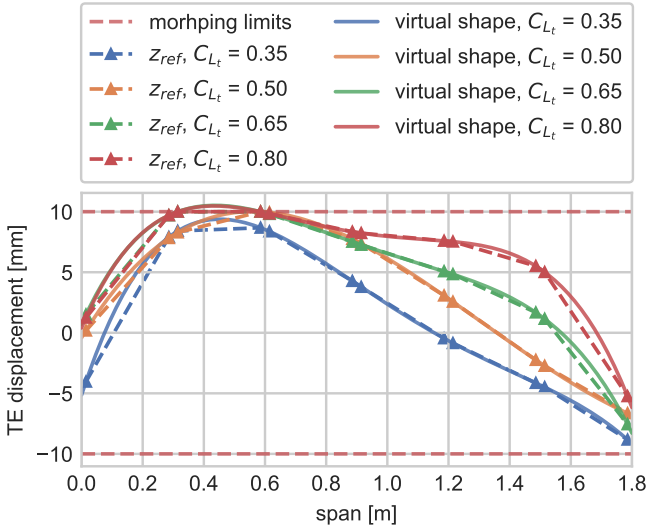


Figure III.3.16: Experimental data-driven optimal shape predictions for four target lift coefficients.

For active camber morphing airfoils, the amount of lift generated can be changed with variations of both the AOA and the airfoil camber. Hence, multiple combinations of the AOA and camber morphing can be employed to realise any given C_{L_t}. However, due to the underlying aerodynamics, these different solutions will not necessarily come with the same drag penalties. Computational Fluid Dynamics (CFD) simulations for active camber morphing airfoils have shown that the trailing-edge deflection for the best lift-to-drag ratio increases with increasing lift coefficients [40]. Moreover, in the case of a three-dimensional distributed active camber morphing wing, C_{L_t} can be realised with different combinations of the AOA and the spanwise camber distribution. Earlier wind tunnel experiments on SmartX-Alpha have shown that for a uniform spanwise camber morphing, the optimal amount of trailing-edge displacement also increases with increasing lift coefficients. Therefore, the trend of increasing the overall camber with increasing C_{L_t} conforms to the expectations.

As long as the morphing limits are not reached, both the best overall amount of camber and the ideal spanwise lift distribution, which is elliptical for induced drag reduction, can be achieved simultaneously. Comparing the camber distributions for $C_{L_t} = 0.35$ and $C_{L_t} = 0.50$, the largest difference between the two distributions is a flat increase of camber along the entire wing span. For the $C_{L_t} = 0.50$ distribution, the 4^{th} and 5^{th} servo units have already reached their upper limits. Consequently, for $C_{L_t} = 0.65$ and $C_{L_t} = 0.80$, no flat camber increases across the span are observed, but rather reshaping of the spanwise distribution occurs. This can be explained by the tradeoff between increasing the overall camber for achieving C_{L_t} and retaining an ideally shaped spanwise lift distribution when the morphing saturation limit is locally reached.

Table III.3.3: Angles of attack and aerodynamic coefficients corresponding to predicted optimal wing shapes.

C_{L_t} [-]	C_D [-]	$\frac{L}{D}$ [-]	α [deg]	C_D reduction [%]
0.35	0.02762	12.67	-2.50	19.8
0.50	0.04188	11.94	-1.05	17.4
0.65	0.06282	10.35	0.81	11.1
0.80	0.09023	8.87	2.97	6.5

The lift and drag coefficients and the angles of attack of the predicted optimal shapes are given in Table III.3.3, which proves that over 6.5 % of drag reduction has been successfully reached for all the tested target lift coefficients. It can be seen from Table III.3.3 that higher C_D reductions are achieved for lower target lift coefficients. As mentioned before, the relatively highly cambered baseline wing shape (NACA6510) is naturally efficient at inducing higher lift coefficients. Consequently, the wing can benefit more from active morphing for lower target lift coefficients. In addition, for a lower C_{L_t}, the associated drag is also lower, meaning that the relative error of the drag coefficient prediction becomes larger, and the maximum lift-to-drag ratio may become optimistic.

Due to experimental time constraints, the measured data set is relatively small but adequate for demonstrating the effectiveness of the proposed optimisation architecture. In the future, with more data, even higher-accuracy neural network models could be trained to enable more accurate evaluations for the highest achievable drag reductions.

Finally, the current optimisation structure allows the implementation and development of other optimisers, which can be implemented with the same black-box data structure in the inner loop, using a similar parallel approach as with the parallel query of CMA-ES candidate solutions.

III.3.7. Conclusions

An online black-box shape optimisation architecture for actively distributed camber morphing wings has been proposed and experimentally validated. Compared to the unmorphed wing base shape, a drag reduction of 7.8 % was achieved on the SmartX-Alpha demonstrator for a target lift coefficient of 0.65. For a wide range of target lift coefficients, the predicted drag reductions vary between 6.5 % and 19.8 %, with higher drag reductions associated with lower lift coefficients.

The ability of the proposed architecture to realise the best wing shape for various lift coefficients online eliminates the need for model excitation manoeuvres at every trim condition, as is characteristic of existing grey-box methods employing local models.

Furthermore, with the inner and outer loops, the online optimisation structure presented in the current Chapter allows the implementation and development of other optimisers, which can be implemented using the same black-box model in the inner loop. It is recommended to investigate Hyperparameter Optimisation (HPO) strategies of the neural network parameters further to enhance the prediction performance of the RBFNN model.

Finally, by virtue of its black-box nature, the proposed optimisation architecture shows promising potential for application to generic morphing wing platforms. Chapter II.2 showed how the RBFNN onboard model could generalise and adapt online to a secondary wing model, similar in configuration but with different aerodynamic baseline shapes.

For future applications to real-world commercial aircraft, the external force-balance, the vision system used in the experimental setup, and the optimisation architecture can be adopted onboard with embedded GPU computational units, such as the Jetson AGX TX2/Xavier and real-time systems target machines such as the Speedgoat [41, 42]. These features make the proposed optimisation architecture effective and practical for achieving sustainable aviation.

References

[1] T. Mkhoyan, O. Ruland, R. De Breuker, and X. Wang, "On-line Black-box Aerodynamic Performance Optimization for a Morphing Wing with Distributed Sensing and Control," *In Review at IEEE Transactions on Control Systems Technology*, 2021.

[2] T. A. Weisshaar, "Morphing aircraft systems: Historical perspectives and future challenges," *Journal of Aircraft*, vol. 50, pp. 337–353, mar 2013.

[3] D. Lentink, U. K. Müller, E. J. Stamhuis, R. De Kat, W. Van Gestel, L. L. Veldhuis, P. Henningsson, A. Hedenström, J. J. Videler, and J. L. Van Leeuwen, "How swifts control their glide performance with morphing wings," *Nature*, vol. 446, no. 7139, pp. 1082–1085, 2007.

[4] M. Jacobsen, "Real time drag minimization using redundant control surfaces," *Aerospace Science and Technology*, vol. 10, no. 7, pp. 574–580, 2006.

[5] Y. Ferrier, N. T. Nguyen, and E. Ting, "Real-Time Adaptive Least-Squares Drag Minimization for Performance Adaptive Aeroelastic Wing," in *34th AIAA Applied Aerodynamics Conference*, (Washington, D.C.), American Institute of Aeronautics and Astronautics, June 2016.

[6] N. Nguyen, N. B. Cramer, K. E. Hashemi, E. Ting, M. Drew, R. Wise, J. Boskovic, N. Precup, T. Mundt, and E. Livne, "Real-Time Adaptive Drag Minimization Wind Tunnel Investigation of a Flexible Wing with Variable Camber Continuous Trailing Edge Flap System," in *AIAA Aviation 2019 Forum*, (Dallas, Texas), p. 3156, AIAA, jun 2019.

[7] N. Nguyen and J. Xiong, "Real-Time Drag Optimization of Aspect Ratio 13.5 Common Research Model with Distributed Flap System," in *AIAA Scitech 2021 Forum*, (virtual event), p. 69, AIAA, January 2021.

[8] J. Snyman, *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*, p. 24. 233 Spring Street, New York, NY 10013, USA: (Springer Science and Business Media, Inc., 2005.

[9] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.

[10] W. Gong, A. Zhou, and Z. Cai, "A Multioperator Search Strategy Based on Cheap Surrogate Models for Evolutionary Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 746–758, 2015.

[11] J. L. Chávez-Hurtado and J. E. Rayas-Sánchez, "Polynomial-Based Surrogate Modeling of RF and Microwave Circuits in Frequency Domain Exploiting the Multinomial Theorem," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 12, pp. 4371–4381, 2016.

[12] W. Zhang, F. Feng, J. Jin, and Q.-J. Zhang, "Parallel Multiphysics Optimization for Microwave Devices Exploiting Neural Network Surrogate," *IEEE Microwave and Wireless Components Letters*, vol. 31, no. 4, pp. 341–344, 2021.

[13] R. G. Regis, "Evolutionary Programming for High-Dimensional Constrained Expensive Black-Box Optimization Using Radial Basis Functions," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326–347, 2014.

[14] B. Liu, Q. Zhang, and G. G. E. Gielen, "A Gaussian Process Surrogate Model Assisted Evolutionary Algorithm for Medium Scale Expensive Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2014.

[15] J. Luo, A. Gupta, Y.-S. Ong, and Z. Wang, "Evolutionary Optimization of Expensive Multiobjective Problems With Co-Sub-Pareto Front Gaussian Process Surrogates," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1708–1721, 2019.

[16] C. S. K. Dash, A. K. Behera, S. Dehuri, and S. B. Cho, "Radial basis function neural networks: A topical state-of-the-art survey," *Open Computer Science*, vol. 6, no. 1, pp. 33–63, 2016.

[17] I. Bajaj, A. Arora, and M. M. F. Hasan, *Black-Box Optimization: Methods and Applications*, pp. 35–65. Cham: Springer International Publishing, 2021.

[18] E. Iuliano and D. Quagliarella, "Efficient aerodynamic optimization of a very light jet aircraft using evolutionary algorithms and Reynolds-averaged Navier–Stokes flow models," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, no. 10, pp. 1109–1129, 2011.

[19] D. Keidel, G. Molinari, and P. Ermanni, "Aero-structural optimization and analysis of a camber-morphing flying wing: Structural and wind tunnel testing," *Journal of Intelligent Material Systems and Structures*, vol. 30, no. 6, pp. 908–923, 2019.

[20] T. Mkhoyan, N. R. Thakrar, R. De Breuker, and J. Sodja, "Design of a Smart Morphing Wing Using Integrated and Distributed Trailing Edge Camber Morphing," in *ASME 2020 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, pp. –, American Society of Mechanical Engineers, sep 2020.

[21] M. Ruderman and L. Fridman, "Model-Free Sliding-Mode-Based Detection and Estimation of Backlash in Drives With Single Encoder," *IEEE Transactions on Control Systems Technology*, vol. 29, pp. 812–817, mar 2021.

[22] X. Wang, T. Mkhoyan, I. Mkhoyan, and R. De Breuker, "Seamless Active Morphing Wing Simultaneous Gust and Maneuver Load Alleviation," *Journal of Guidance, Control, and Dynamics*, vol. 44, pp. 1649–1662, sep 2021.

[23] D. Papageorgiou, M. Blanke, H. H. Niemann, and J. H. Richter, "Robust Backlash Estimation for Industrial Drive-Train Systems—Theory and Validation," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 5, pp. 1847–1861, 2019.

[24] P. Serra, R. Cunha, T. Hamel, C. Silvestre, and F. Le Bras, "Nonlinear image-based visual servo controller for the flare maneuver of fixed-wing aircraft using optical flow," *IEEE Transactions on Control Systems Technology*, vol. 23, pp. 570–583, mar 2015.

[25] Y. Xu, S. Yin, S. X. Ding, H. Luo, and Z. Zhao, "Performance Degradation Monitoring and Recovery of Vision-Based Control Systems," *IEEE Transactions on Control Systems Technology*, vol. 29, pp. 2712–2719, nov 2021.

[26] M. M. Gomroki, F. Topputo, F. Bernelli-Zazzera, and O. Tekinalp, "Solving Constrained Optimal Control Problems Using State-Dependent Factorization and Chebyshev Polynomials," *Journal of Guidance, Control, and Dynamics*, vol. 41, pp. 618–631, mar 2018.

[27] N. Hansen and A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies," *Evolutionary Computation*, vol. 9, pp. 159–195, June 2001.

[28] N. Hansen, *The CMA Evolution Strategy: A Comparing Review*, pp. 75–102. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[29] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.

[30] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2 ed., Mar. 2004.

[31] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive Real-Time Clustering Method for Dynamic Visual Tracking of Very Flexible Wings," *Journal of Aerospace Information Systems*, vol. 18, pp. 58–79, jan 2021.

[32] K. Kanatani, Y. Sugaya, and Y. Kanazawa, "Multiview Triangulation," in *Guide to 3D Vision Computation*, pp. 133–147, Cham: Springer International Publishing, 2016. Series Title: Advances in Computer Vision and Pattern Recognition.

[33] R. Hartley and F. Kahl, "Optimal Algorithms in Multiview Geometry," in *Computer Vision – ACCV 2007*, vol. 4843, (Berlin, Heidelberg), pp. 13–34, Springer Berlin Heidelberg, 2007. Series Title: Lecture Notes in Computer Science.

[34] OptiTrack, "OptiTrack - Primex 41 - In Depth."

[35] B. Sun and E. J. van Kampen, "Intelligent adaptive optimal control using incremental model-based global dual heuristic programming subject to partial observability," *Applied Soft Computing*, vol. 103, no. February, 2021.

[36] E. Tal and S. Karaman, "Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness," *IEEE Transactions on Control Systems Technology*, vol. 29, pp. 1203–1218, may 2021.

[37] J.-W. van Wingerden, A. Hulskamp, T. Barlas, I. Houtzager, H. Bersee, G. van Kuik, and M. Verhaegen, "Two-Degree-of-Freedom Active Vibration Control of a Prototyped "Smart" Rotor," *IEEE Transactions on Control Systems Technology*, vol. 19, pp. 284–296, mar 2011.

[38] multiSIM, "multiSIM: Distributed simulation D-SIM," 2021.

[39] D. Isele and A. Cosgun, "Selective Experience Replay for Lifelong Learning," in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, (New Orleans, Louisiana), pp. 3302 – 3309, American Institute of Aeronautics and Astronautics (AIAA), 2018.

[40] S. J. Huntley, B. K. Woods, and C. B. Allen, "Computational Analysis of the Aerodynamics of Camber Morphing," in *AIAA Aviation 2019 Forum*, (Dallas, Texas), American Institute of Aeronautics and Astronautics, June 2019.

[41] NVIDIA, "High Performance AI at the Edge | NVIDIA Jetson TX2."

[42] Speedgoat, "Performance real-time target machine | Speedgoat."

Appendices

III.A1

Appendices Smart Control

III.A1.1. Modal Testing

To support the identification of the structural and aerodynamic parameters of the morphing wing, a Ground Vibration Test (GVT) was performed on the fixed-free clamped (cantilever) wing. The following analysis is presented as supporting data for the wind tunnel campaign described in Chapter III.1 and in publication [1].

The modal testing campaign was realised in the Open Jet Facility (OJF) at the Delft University of Technology alongside the wing tunnel experiment performed in Chapter III.1. A GVT setup was constructed where the wing was subjected to a known excitation (input) using a modal shaker. The response (output) was subsequently measured using a laser vibrometer system.

III.A1.1.1. Experimental Setup

The experimental hardware, the type of excitation used in the structural vibration tests, and data acquisition are described in this section.

Hardware

The schematic of the test setup is shown in Fig. III.A1.1. The wing was mounted on a turntable with a force-balance system, which in turn was attached to a heavy-duty cast iron test table. A 31 N modal shaker with a built-in amplifier (K2007E01 Smart Shaker [2]) was used for excitation during testing. The shaker was secured on a separate solid base such that the construction would not touch the test table. To connect the shaker to the driving point on the wing a long (50 cm) flexible stinger was manufactured. The driving point was chosen near the leading edge on the bottom surface of the wing and approximately 20 cm above the root. A mechanical impedance sensor, attached to the stinger-wing interface, measured the force and the acceleration at the driving point with a sensitivity and broadband resolution of 10.2 mV/m/s^2 and 0.02 m/s^2 rms, and 22.4 mV/N and 0.0089 N in acceleration and force respectively. The force and acceleration signals from the impedance sensor were fed to the PSV-500 data acquisition system as reference signals.

III.A1

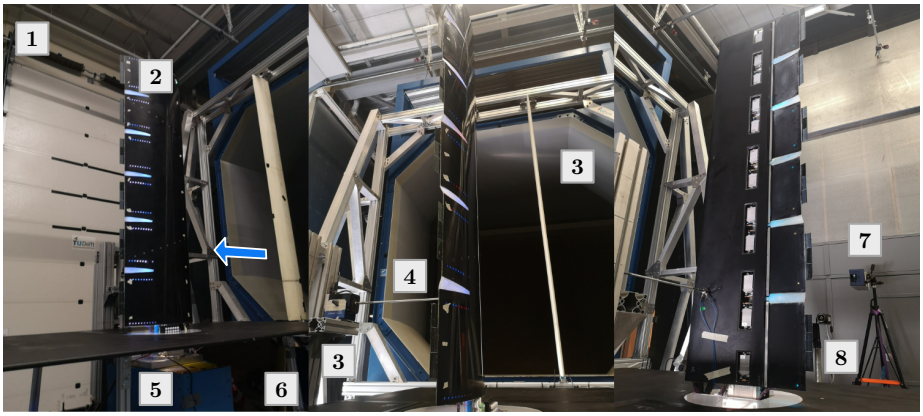


Figure III.A1.1: Schematic of the Ground Vibration Testing (GVT) setup: 1) SmartX-Alpha wing; 2) Open-Jet Facility (OJF) subsonic wind tunnel test section; 3) modal shaker; 4) modal stinger; 5) turntable with force balance system; 6) heavy-duty cast iron test table; 7) Polytec PSV-500-1D laser vibrometer; 8) PSV DAQ PC.

Excitation Input

Sine sweep excitation was used for all runs of measurements, structural and aeroelastic. Compared to harmonic excitation at a distinct frequency, the sine sweep significantly reduces the measurement time yet still provides a good Signal-to-Noise Ratio (SNR) [3, 4].

Sine sweep excitation was used for all runs of measurements. The signal is represented by the following equation:

$$F_{\text{sweep}}(t) = A \sin\left(2\pi f_1 t + 2\pi \frac{f_2 - f_1}{T_{tot}} t^2\right) \tag{III.A1.1}$$

where f_1 and f_2 are the initial and final frequencies of the sine sweep signal in Hz, A is the wave amplitude, and T_{tot} is the total measurement time.

Data Acquisition

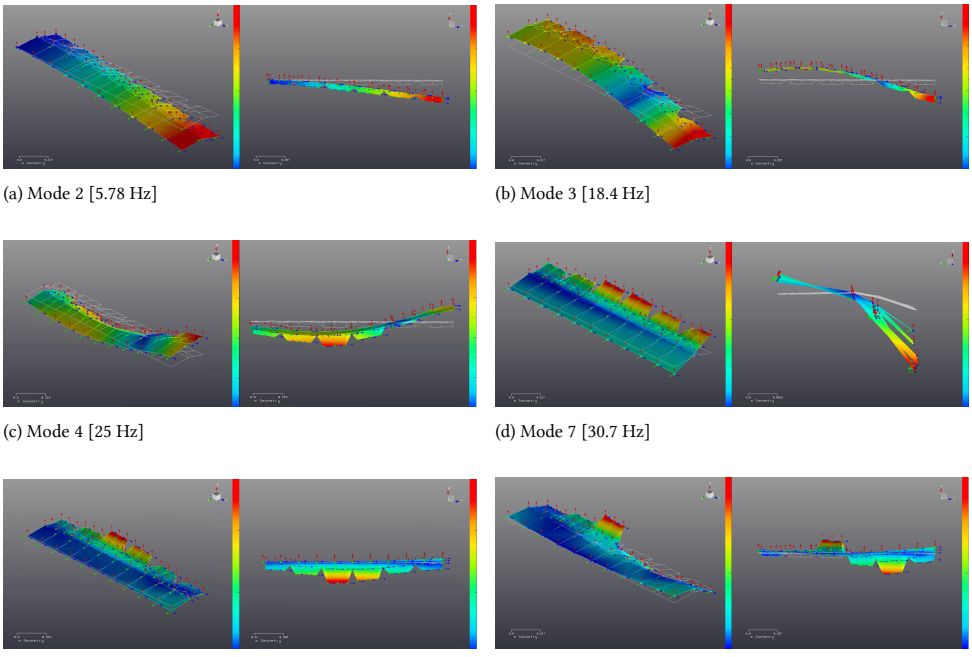
Similar to the setup of Chapter I.2, a Polytec PSV-500 laser vibrometer system [4] with a resolution (RMS) of 200 µm/s was used to measure the vibration response of the wing in terms of the velocity and displacement. Since the laser allowed for the measurement of only a single point at a time, to be able to reconstruct the vibrational shape of the wing, multiple measurements had to be taken along the surface of the vibrating wing. To ensure the quality of those measurements, multiple measurements had to be taken at each scan point and averaged across the corresponding re-runs. For all vibrometry measurements, 58 scan points were distributed along the upper surface of the wing, of which 22 scan points along the main wing body and 2×18 evenly spread over the 6 actuator modules along two rows. Each scan point measurement, of approximately 30 seconds, was repeated 15 times and the vibration signal averaged over those repeated runs. The frequency range of interest was 0-100 Hz; a sampling rate of 5 times the maximum range was used.

III.A1.1.2. Modal Analysis

The postprocessing and modal analysis of the acquired vibration data was performed using the Simcenter Testlab 2021 software [5]. The result of the GVT yielded a set of structural modes as shown in Fig. III.A1.2. The main first and second wing bending were found at 5.8 and 18.5 Hz, respectively. The general torsional mode was found at 30.7 Hz. The remaining modes in Fig. III.A1.2 are coupled with control surface modes. Since SmartX-Alpha has 6 distributed flaps, there are several symmetric and asymmetric control panel modes coupled with wing bending.

Table III.A1.1: SmartX-Alpha Identified structural modes from vibration tests.

Mode	N.Freq., f [Hz]	Damping, η [%]	Description
1	4.90	1.22	Coupled Rigid Body Mode – base/wing root translation in UZ + wing rotation RX
2	5.76	0.7	1st global wing bending in RX (longitudinal) – fixed-free wing with flaps
3	18.4	0.9	2ND global wing bending in RX (longitudinal), with base translation in UZ
4/5	24.9/26.5	0.8/1.16	Coupled mode global longitudinal (RX) / lateral (RY) bending
6	27.6	1.22	Global wing bending with (inboard) control panel modes
7	30.7	2.2	1st Wing Torsion
8-15	34-60	-	Control panel modes



(a) Mode 2 [5.78 Hz]

(b) Mode 3 [18.4 Hz]

(c) Mode 4 [25 Hz]

(d) Mode 7 [30.7 Hz]

(e) Mode 8 [34 Hz]

(f) Mode 10 [44.7 Hz]

Figure III.A1.2: SmartX-Alpha main modes 0-60 Hz.

The modal analysis results in Fig. III.A1.2 reveal imperfect fixed-free (one end clamped) boundary conditions of this experimental setup. The heavy-duty cast iron test table (refer to Fig. III.A1.1-(6)) –on top of which the turntable/force-balance holding the wing was mounted– exhibited a natural frequency at 4.9 Hz (Mode 1). Moreover, as the clamped end is not perfectly fixed, the support structure essentially acts as an extension of the wing and contributes to its dynamic behaviour. As no reference measurement was taken of any point on the test table or the clamp directly during the laser vibrometer runs, the data needed to eliminate the effect of the support on the wing dynamics in the postprocessing step was not directly available. Therefore, the presented mode shapes exhibit slight translations or rotations at the clamped end of the wing w.r.t. the undeformed shape.

It should be noted that the purpose of the vibration test was not to update a FEM model using these modal results; if so, then the test table, the turntable, and the clamp – the entire support would have to be modelled along with the wing itself. In contrast, the objective of the performed tests was to identify the properties of the very structural system that would be subjected to subsequent wind tunnel tests. This structural system is not the wing on its own but includes every structural component supporting it during the test run. As such, there was a practical need to retain the same real boundary conditions on the structure as those existing during the wind tunnel experiments.

III.A1.2. Proof of Theorems

Below, the proof of theorems in Chapter III.1 and [6] are presented.

Proof of Theorem 1:

Denote the initial time point as t_*. Choose $V_1(\boldsymbol{e}) = \boldsymbol{e}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{e}$, where $\boldsymbol{P} = \boldsymbol{P}^{\mathrm{T}} > 0$ is the solution of the Lyapunov equation $\boldsymbol{P}(\boldsymbol{A}_c - \boldsymbol{B}_c\boldsymbol{K}) + (\boldsymbol{A}_c - \boldsymbol{B}_c\boldsymbol{K})^{\mathrm{T}}\boldsymbol{P} = -\boldsymbol{I}$. Then $\alpha_1(\|\boldsymbol{e}\|_2) \leq V_1(\boldsymbol{e}) \leq \alpha_2(\|\boldsymbol{e}\|_2)$, $\alpha_1(\|\boldsymbol{e}\|_2) \triangleq \lambda_{\min}(\boldsymbol{P})\|\boldsymbol{e}\|_2^2$, $\alpha_2(\|\boldsymbol{e}\|_2) \triangleq \lambda_{\max}(\boldsymbol{P})\|\boldsymbol{e}\|_2^2$. α_1, α_2 belong to the class $\mathcal{K}_\infty$ functions. Using Eq. (III.1.5), the time derivative V_1 is:

$$\begin{aligned}\dot{V}_1 =& \boldsymbol{e}^{\mathrm{T}}[\boldsymbol{P}(\boldsymbol{A}_c - \boldsymbol{B}_c\boldsymbol{K}) + (\boldsymbol{A}_c - \boldsymbol{B}_c\boldsymbol{K})^{\mathrm{T}}\boldsymbol{P}]\boldsymbol{e} + 2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{B}_c\boldsymbol{\varepsilon}_{\text{indi}} \\ \leq& -\|\boldsymbol{e}\|_2^2 + 2\|\boldsymbol{e}\|_2\|\boldsymbol{P}\boldsymbol{B}_c\|_2\bar{\varepsilon} \leq -\theta_1\|\boldsymbol{e}\|_2^2, \quad \forall\|\boldsymbol{e}\|_2 \geq \frac{2\|\boldsymbol{P}\boldsymbol{B}_c\|_2\bar{\varepsilon}}{1-\theta_1} \triangleq \mu_1\bar{\varepsilon}\end{aligned} \tag{III.A1.2}$$

with constant $\theta_1 \in (0,1)$. Consequently, for $\forall\, \boldsymbol{e}(t_*) \in \mathbb{R}^\rho$, there exists a class $\mathcal{KL}$ function β and finite $T_1 \geq 0$ such that $\|\boldsymbol{e}(t)\|_2 \leq \beta(\|\boldsymbol{e}(t_*)\|_2, t-t_*)$, $t_* \leq \forall\, t \leq t_*+T_1$, $\|\boldsymbol{e}(t)\|_2 \leq \alpha_1^{-1}(\alpha_2(\mu_1\bar{\varepsilon}))$, $\forall\, t \geq t_* + T_1 \triangleq t'_*$. In other words, the tracking error $\boldsymbol{e}$ is bounded for all $t \geq t_*$ and is ultimately bounded by $\alpha_1^{-1}(\alpha_2(\mu_1\bar{\varepsilon})) = \sqrt{\lambda_{\max}(\boldsymbol{P})/\lambda_{\min}(\boldsymbol{P})}\mu_1\bar{\varepsilon}$.

Regarding the internal dynamics, because the origin of $\dot{\boldsymbol{\eta}} = \boldsymbol{f}_\eta(\boldsymbol{\eta}, \boldsymbol{0}, \boldsymbol{0})$ is globally exponentially stable, then there exists a Lyapunov function $V_2(\boldsymbol{\eta})$ defined in $D_\eta = \{\boldsymbol{\eta} \in \mathbb{R}^{n-\rho}\}$ that satisfies $c_1\|\boldsymbol{\eta}\|_2^2 \leq V_2(\boldsymbol{\eta}) \leq c_2\|\boldsymbol{\eta}\|_2^2$, $\frac{\partial V_2}{\partial \boldsymbol{\eta}}\boldsymbol{f}_\eta(\boldsymbol{\eta}, \boldsymbol{0}, \boldsymbol{0}) \leq -c_3\|\boldsymbol{\eta}\|_2^2$, $\left\|\frac{\partial V_2}{\partial \boldsymbol{\eta}}\right\|_2 \leq c_4\|\boldsymbol{\eta}\|_2$, for some positive constants c_1, c_2, c_3, c_4. Denote $\alpha'_1(\|\boldsymbol{\eta}\|_2) \triangleq c_1\|\boldsymbol{\eta}\|_2^2, \alpha'_2(\|\boldsymbol{\eta}\|_2) \triangleq c_2\|\boldsymbol{\eta}\|_2^2$, then α'_1, α'_2 belong to class $\mathcal{K}_\infty$ functions. Furthermore, because $\boldsymbol{f}_\eta(\boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{d})$ is continuously differentiable and globally Lipschitz in $(\boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{d})$, then there exists a global Lipschitz constant L such that $\|\boldsymbol{f}_\eta(\boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{d}) - \boldsymbol{f}_\eta(\boldsymbol{\eta}, \boldsymbol{0}, \boldsymbol{0})\|_2 \leq L(\|\boldsymbol{e}\|_2 + \|\boldsymbol{\mathcal{R}}\|_2 + \|\boldsymbol{d}\|_2)$, $\forall \boldsymbol{\eta} \in \mathbb{R}^{n-\rho}$. As a result, the time derivative of $V_2(\boldsymbol{\eta})$ satisfies:

$$\begin{aligned}\dot{V}_2(\boldsymbol{\eta}) =& \frac{\partial V_2}{\partial \boldsymbol{\eta}}\boldsymbol{f}_\eta(\boldsymbol{\eta}, \boldsymbol{\xi}, \boldsymbol{d}) \leq -c_3\|\boldsymbol{\eta}\|_2^2 + c_4L\|\boldsymbol{\eta}\|_2(\|\boldsymbol{e}\|_2 + \bar{\mathcal{R}} + \bar{d}) \leq -c_3(1-\theta_2)\|\boldsymbol{\eta}\|_2^2, \ \forall\|\boldsymbol{\eta}\|_2 \\ \geq& \frac{c_4L(\|\boldsymbol{e}\|_2 + \bar{\mathcal{R}} + \bar{d})}{c_3\theta_2}\end{aligned} \tag{III.A1.3}$$

with constant $\theta_2 \in (0,1)$. Denote

$$\mu_2 \triangleq c_4L(\sup_{t'_* \leq \tau \leq t}\|\boldsymbol{e}\|_2 + \bar{\mathcal{R}} + \bar{d})/(c_3\theta_2) \triangleq \theta_3(\sup_{t'_* \leq \tau \leq t}\|\boldsymbol{e}\|_2 + \bar{\mathcal{R}} + \bar{d}) \tag{III.A1.4}$$

then $\dot{V}_2(\boldsymbol{\eta}) \leq -c_3(1-\theta_2)\|\boldsymbol{\eta}\|_2^2$, $\forall\|\boldsymbol{\eta}\|_2 \geq \mu_2$, $\forall t \geq t'_*$. Consequently, there exists a class $\mathcal{KL}$ function β' such that $\|\boldsymbol{\eta}(t)\|_2 \leq \beta'(\|\boldsymbol{\eta}(t'_*)\|_2, t-t'_*) + \alpha_1'^{-1}(\alpha'_2(\mu_2))$, $\forall t \geq t'_*$. Since β' is a $\mathcal{KL}$ function, then the norm value of $\boldsymbol{\eta}(t)$ yields $\|\boldsymbol{\eta}(t)\|_2 \leq \theta_4\bar{\varepsilon} + \alpha_1'^{-1}(\alpha'_2(\theta_3(\alpha_1^{-1}(\alpha_2(\mu_1\bar{\varepsilon})) + \bar{\mathcal{R}} + \bar{d})))$, $\forall t \geq t_* + T_1 + T_2$ for some finite $T_2 > 0$ and $\theta_4 > 0$. In other words, $\boldsymbol{\eta}$ is globally ultimately bounded by a class $\mathcal{K}$ function of $\bar{\varepsilon}$, $\bar{\mathcal{R}}$, and $\bar{d}$. □

Proof of Theorem 2:

Essentially, Theorem 2 is a local version of Theorem 1. When global Lipschitz and global exponential stability are not ensured, the stability criteria impose constraints on both initial conditions and perturbation bound. Because the conditions for tracking error remain unchanged, Eq. (III.A1.2) still holds, which proves that $\boldsymbol{e}$ is ultimately bounded

III.A1

by a class $\mathcal{K}$ function of $\bar{\varepsilon}$. Nevertheless, a $V_2(\boldsymbol{\eta})$ and a Lipschitz constant only exist in a neighborhood of $\boldsymbol{\eta} = \mathbf{0}$, which is denoted as $D'_\eta = \{\boldsymbol{\eta} \in \mathbb{R}^{n-\rho} | \ \|\boldsymbol{\eta}\|_2 < r_\eta\}$. Take $0 < r < r_\eta$ such that $D_r \subset D'_\eta$. According to the boundedness theories [7], Eq. (III.A1.3) only holds when $\mu_2 < \alpha_2'^{-1}(\alpha'_1(r))$, $\|\boldsymbol{\eta}(t'_*)\|_2 \leq \alpha_2'^{-1}(\alpha'_1(r))$. Using Eq. (III.A1.4), the perturbation is constrained by $\bar{\varepsilon} < \varepsilon^* \triangleq (1/\mu_1)\alpha_2^{-1}(\alpha_1((1/\theta_3)(\alpha_2'^{-1}(\alpha'_1(r))) - \bar{\mathcal{R}} - \bar{d}))$. When the constraints on the initial condition and perturbation bound are satisfied, $\boldsymbol{\eta}$ is ultimately bounded by a class $\mathcal{K}$ function of $\bar{\varepsilon}$, $\bar{\mathcal{R}}$, and $\bar{d}$. □

Proof of Theorem 3: Recall Eqs. (III.1.3, III.1.4, III.1.5), the output dynamics under INDI control can also be written as $\boldsymbol{y}^{(\boldsymbol{\rho})} = \boldsymbol{\nu}_c + \boldsymbol{\varepsilon}_{\text{indi}}$. Also, at the previous time step, $\boldsymbol{y}_0^{(\boldsymbol{\rho})} = \boldsymbol{\nu}_{c_0} + \boldsymbol{\varepsilon}_{\text{indi}_0}$. Therefore, using Eq. (III.1.5), $\boldsymbol{\varepsilon}_{\text{indi}}$ can be rewritten as

$$\begin{aligned}\boldsymbol{\varepsilon}_{\text{indi}} =&(\boldsymbol{\mathcal{B}}(\boldsymbol{x}_0)\bar{\boldsymbol{\mathcal{B}}}^+(\boldsymbol{x}_0) - \boldsymbol{I}_{p\times p})(\boldsymbol{\nu}_c - \boldsymbol{y}_0^{(\boldsymbol{\rho})}) + \boldsymbol{\delta}(\boldsymbol{x}, \Delta t) + \boldsymbol{\varepsilon}_{\text{ca}} + \Delta\boldsymbol{d}_y \\ =&(\boldsymbol{I}_{p\times p} - \boldsymbol{\mathcal{B}}(\boldsymbol{x}_0)\bar{\boldsymbol{\mathcal{B}}}^+(\boldsymbol{x}_0))\boldsymbol{\varepsilon}_{\text{indi}_0} - (\boldsymbol{I}_{p\times p} - \boldsymbol{\mathcal{B}}(\boldsymbol{x}_0)\bar{\boldsymbol{\mathcal{B}}}^+(\boldsymbol{x}_0))(\boldsymbol{\nu}_c - \boldsymbol{\nu}_{c_0}) + \boldsymbol{\delta}(\boldsymbol{x}, \Delta t) + \boldsymbol{\varepsilon}_{\text{ca}} + \Delta\boldsymbol{d}_y \\ \triangleq&\boldsymbol{E}\boldsymbol{\varepsilon}_{\text{indi}_0} - \boldsymbol{E}\Delta\boldsymbol{\nu}_c + \boldsymbol{\delta}(\boldsymbol{x}, \Delta t) + \boldsymbol{\varepsilon}_{\text{ca}} + \Delta\boldsymbol{d}_y \end{aligned} \tag{III.A1.5}$$

which can be written in a recursive way as $\boldsymbol{\varepsilon}_{\text{indi}}(k) = \boldsymbol{E}(k)\boldsymbol{\varepsilon}_{\text{indi}}(k-1) - \boldsymbol{E}(k)\Delta\boldsymbol{\nu}_c(k) + \boldsymbol{\delta}(k) + \boldsymbol{\varepsilon}_{\text{ca}}(k) + \Delta\boldsymbol{d}_y(k)$. When the input constraints are not considered, the control allocation error $\boldsymbol{\varepsilon}_{\text{ca}}$ equals zero. Moreover, $\boldsymbol{\nu}_c$ is designed to be continuous in time (Eq. (III.1.4)), thus $\lim_{\Delta t\to 0} \|\boldsymbol{\nu}_c - \boldsymbol{\nu}_{c_0}\|_2 = 0$, $\forall \boldsymbol{x} \in \mathbb{R}^n$. This equation also indicates that $\forall\ \overline{\Delta\nu}_c > 0$, $\exists\ \overline{\Delta t} > 0, s.t.$ for all $0 < \Delta t \leq \overline{\Delta t}$, $\forall \boldsymbol{x} \in \mathbb{R}^n$, $\|\boldsymbol{\nu}_c - \boldsymbol{\nu}_{c_0}\|_2 \leq \overline{\Delta\nu}_c$. As a consequence, the following equation holds:

$$\begin{aligned}\|\boldsymbol{\varepsilon}_{\text{indi}}(k)\|_2 \leq&(\bar{b})^k\|\boldsymbol{\varepsilon}_{\text{indi}}(t=0)\|_2 + \sum_{j=1}^{k}(\bar{b})^{k-j+1}\|\Delta\boldsymbol{\nu}_c(j)\|_2 \\ &+ \sum_{j=1}^{k-1}(\bar{b})^{k-j}\|\boldsymbol{\delta}(j) + \Delta\boldsymbol{d}(j)\|_2 + \|\boldsymbol{\delta}(k) + \Delta\boldsymbol{d}(k)\|_2 \\ \leq&(\bar{b})^k\|\boldsymbol{\varepsilon}_{\text{indi}}(t=0)\|_2 + \overline{\Delta\nu}_c\frac{\bar{b} - \bar{b}^{k+1}}{1-\bar{b}} + (\bar{\delta} + \overline{\Delta d})\frac{1-\bar{b}^k}{1-\bar{b}}\end{aligned} \tag{III.A1.6}$$

Because $\bar{b} < 1$, then $\|\boldsymbol{\varepsilon}_{\text{indi}}\|_2 \leq \frac{\overline{\Delta\nu}_c\bar{b}+\bar{\delta}+\overline{\Delta d}}{1-\bar{b}}$, as $k \to \infty$. In conclusion, $\boldsymbol{\varepsilon}_{\text{indi}}$ is ultimately bounded by $\frac{\overline{\Delta\nu}_c\bar{b}+\bar{\delta}+\overline{\Delta d}}{1-\bar{b}}$. □

Proof of Theorem 4:

In contrast to Eq. (III.1.7), the analytical expression for the control increment given by quadratic programming $\Delta\boldsymbol{u}_{\text{indi-qp}}$ is unknown. Instead, the only information about $\Delta\boldsymbol{u}_{\text{indi-qp}}$ is that it satisfies $\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\Delta\boldsymbol{u}_{\text{indi-qp}} = \boldsymbol{\nu}_c - \boldsymbol{y}_0^{(\boldsymbol{\rho})} + \boldsymbol{\varepsilon}_{\text{ca}}$, where $\boldsymbol{\varepsilon}_{\text{ca}}$ is the control allocation error. Using Eqs. (III.1.3, III.1.4, III.1.5), the corresponding $\boldsymbol{\varepsilon}_{\text{indi}}$ is derived as

$$\begin{aligned}\boldsymbol{\varepsilon}_{\text{indi}} =&(\boldsymbol{K}_{\mathcal{B}}(\boldsymbol{x}_0) - \boldsymbol{I}_{p\times p})(\boldsymbol{\nu}_c - \boldsymbol{\nu}_{c_0} - \boldsymbol{\varepsilon}_{\text{indi}_0} + \boldsymbol{\varepsilon}_{\text{ca}}) + \boldsymbol{\delta}(\boldsymbol{x}, \Delta t) + \boldsymbol{\varepsilon}_{\text{ca}} \\ &+\Delta\boldsymbol{d}_y \triangleq \boldsymbol{E}'\boldsymbol{\varepsilon}_{\text{indi}_0} - \boldsymbol{E}'(\Delta\boldsymbol{\nu}_c + \boldsymbol{\varepsilon}_{\text{ca}}) + \boldsymbol{\delta}(\boldsymbol{x}, \Delta t) + \boldsymbol{\varepsilon}_{\text{ca}} + \Delta\boldsymbol{d}_y\end{aligned} \tag{III.A1.7}$$

which can be written in a recursive way as $\boldsymbol{\varepsilon}_{\text{indi}}(k) = \boldsymbol{E}'(k)\boldsymbol{\varepsilon}_{\text{indi}}(k-1) - \boldsymbol{E}'(k)(\Delta\boldsymbol{\nu}_c(k) + \boldsymbol{\varepsilon}_{\text{ca}}(k)) + \boldsymbol{\delta}(k) + \boldsymbol{\varepsilon}_{\text{ca}}(k) + \Delta\boldsymbol{d}_y(k)$. Analogous to the proof of Theorem 3, given a non-zero but bounded $\boldsymbol{\varepsilon}_{\text{ca}}$, the resulting $\boldsymbol{\varepsilon}_{\text{indi}}$ is bounded for all k, and is ultimately bounded by $\frac{\overline{\Delta\nu}_c\bar{b}'+\bar{\delta}+\overline{\Delta d}+(\bar{b}'+1)\bar{\varepsilon}_{\text{ca}}}{1-\bar{b}'}$. □

Proof of Corollary 1:

Denote the solution of the INDI control with quadratic programming control allocation considering virtual shapes as $\Delta \boldsymbol{u}_{\text{indi-qp-v}}$, then correspondingly, $\boldsymbol{\varepsilon}_{\text{indi}} = \boldsymbol{\delta}(\boldsymbol{x}, \Delta t) + (\boldsymbol{\mathcal{B}}(\boldsymbol{x}_0)\boldsymbol{\Phi}_{\bar{x}_s} - \bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\boldsymbol{\Phi}_{\bar{x}_s})\Delta \boldsymbol{u}_{\text{indi-qp-v}} + \boldsymbol{\varepsilon}_{\text{ca}} + \Delta \boldsymbol{d}_y$. It is known that the control allocation leads to
$\left(\bar{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\boldsymbol{\Phi}_{\bar{x}_s}\right) \Delta \boldsymbol{u}_{\text{indi-qp-v}} = \boldsymbol{\nu}_c - \boldsymbol{y}_0^{(\boldsymbol{\rho})} + \boldsymbol{\varepsilon}_{\text{ca}}$, thus:

$$\boldsymbol{\varepsilon}_{\text{indi}} = (\boldsymbol{K}_{\mathcal{B}}(\boldsymbol{x}_0) - \boldsymbol{I}_{p\times p})(\boldsymbol{\nu}_c - \boldsymbol{y}_0^{(\boldsymbol{\rho})} + \boldsymbol{\varepsilon}_{\text{ca}}) + \boldsymbol{\delta}(\boldsymbol{x}, \Delta t) + \boldsymbol{\varepsilon}_{\text{ca}} + \Delta \boldsymbol{d}_y \tag{III.A1.8}$$

Analogous to the proof of Theorem 4, $\boldsymbol{\varepsilon}_{\text{indi}}$ is ultimately bounded by $\frac{\overline{\Delta\nu}_c \bar{b}' + \bar{\delta} + \overline{\Delta d} + (\bar{b}'+1)\bar{\varepsilon}_{\text{ca}}}{1-\bar{b}'}$. □

III.3

References

[1] T. Mkhoyan, V. L. Stuber, X. Wang, I. Mkhoyan, R. De Breuker, and S. van der Zwaag, "Bio-inspired Distributed and Decentralized Real-time Control and Sensing Architecture for a Seamless Active Morphing Wing," *Engineering Journal*, 2022, to be submitted.

[2] Modal Shop, "Modal Shop Smart Shaker with Integrated Power Amplifier," 2021.

[3] D. J. Ewins, *Modal Testing 2: Theory, Practice and Application.* Research Studies Press, 2 ed.

[4] Polytec, "Polytec Single-Point Vibrometers," 2020.

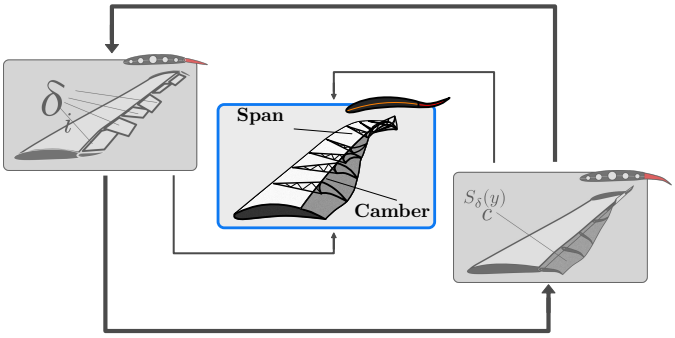
[5] PLM Siemens, "Simcenter Testlab," 2021.

[6] X. Wang, T. Mkhoyan, I. Mkhoyan, and R. De Breuker, "Seamless Active Morphing Wing Simultaneous Gust and Maneuver Load Alleviation," *Journal of Guidance, Control, and Dynamics*, pp. 1649–1662, 2021.

[7] H. K. Khalil, *Nonlinear Systems.* New Jersey: Prentice-Hall, 2002.

IV

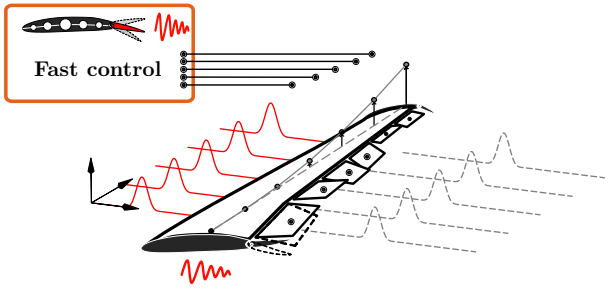
Outlook



IV.1

Simplifications: Distributed and Discretised Morphing

In previous Parts of this dissertation, the development, realisation and validation of an active morphing wing were presented, demonstrating the achievement of multiple control objectives through smart active morphing design. However, actuator bandwidth and mechanical imperfections were identified as areas of improvement associated with the complexities of morphing wing designs. This Chapter addresses those areas and concludes the work of this dissertation through the evaluation and realisation of a distributed and over-actuated aeroelastic wing demonstrator, with the assessment of faster actuation and simplified mechanical design. Aerodynamic and structural analyses were performed to determine actuator torque requirements and actuation mechanism design. The effect of actuator design was studied through a series of gust simulations of closed-loop control of a parametric aeroservoelastic model for gusts with various frequencies [1].



SmartX-Neo

This Chapter is based on the following peer-reviewed conference and journal paper:
T. Mkhoyan, X. Wang, and R. De Breuker, "Aeroelastic Wing Demonstrator with a Distributed and Decentralized Control Architecture," *AIAA SciTech Forum 2022; Submitted to AIAA Journal*, p. 1551, 2022

IV.1

This Chapter investigates the design and development of an autonomous aeroservoelastic wing concept with distributed flaps. This wing demonstrator was developed in the scope of the SmartX project, aiming to demonstrate in-flight performance optimisation and multi-objective control with over-actuated wing designs. Following a successful test campaign with a previous wing design based on active morphing, this study aims to develop an over-actuated aeroelastic wing design suitable for aeroelastic control, including flutter suppression, manoeuvre and gust load alleviation. A decentralised control architecture is developed for the over-actuated and over-sensed system, allowing efficient sensing data processing and control algorithms. Aerodynamic and structural analyses are performed to determine actuator torque requirements and actuation mechanism design. Furthermore, buckling analysis is performed to size the wing structure. A state-space aeroelastic dynamic model is established to analyse the gust response and control effectiveness of the wing. It is established that a linear quadratic regulator significantly improves the closed-loop performance. Furthermore, the hypotheses are confirmed that fast actuation improves load alleviation performance and high-frequency disturbance rejection effectiveness. The manufacturing and integration of the wing demonstrator are discussed, which lay a foundation for future static and dynamic wind-tunnel experiments.

IV.1.1. Introduction

Advancements in aircraft materials, manufacturing technology, control algorithms, and hardware design enable the development of increasingly flexible aircraft concepts. Generally, flexibility comes as a side effect of lighter aircraft design and must be adequately considered in the design.

However, a more natural approach is to utilise the flexibility for the benefit of better performance, much like it is seen in nature with wing morphing for better gliding performance [2, 3]. As in nature, flexible wing concepts have been evolving since the early years of aviation. One of the well-documented examples was the active roll control of the Wright Flyer, the first successful heavier-than-air powered aircraft. In this lightweight design, the lateral stability was ensured by wing twist-warping [4]. This was possible because the flexible fabric-wrapped structure was well suited for morphing.

As the flight speeds and loads increased with the advancement of flight, a stiffer wing was required to fulfil the structural requirements and overcome aeroelastic instabilities. As a result, the considerably more rigid wing design - generally optimised for cruise conditions - is faced with a compromised performance under other flight conditions. To harness the potential of a flexible wing, two design choices are possible: active morphing design and conventionally flapped distributed wing designs. Both design concepts can allow the lift distribution to be tailored actively, potentially reducing this performance loss and improving aircraft performance across the flight envelope. Furthermore, both design concepts can be distributed and modular (i.e. having multiple flaps along the span). Both concepts mimic the distributed nature of feathers found in avian biology.

While active morphing benefits aerodynamic efficiency, the morphing mechanism required for smooth shape control generally needs larger actuation forces and a more complex design. In our previous study, we demonstrated a seamless morphing wing concept [5, 6], the SmartX-Alpha, capable of performing objectives such as shape control, drag minimisation, and simultaneous gust and manoeuvre load alleviation [7]. This design showed a significant advantage over previous morphing concepts, allowing the lift distribution to be controlled locally by individually adjusting the camber and twist of each

morphing module. However, the complexity of the morphing mechanism and increased torque required for morphing demand actuators with high continuous torque. The current study aims to address this gap and investigate the potential of *discrete morphing* with conventionally free hinged flaps. The benefit is significantly lower actuation forces and a simpler actuation mechanism. This objective initiated the development of the SmartX-Neo wing demonstrator concept.

This Chapter describes the design and aeroelastic analysis of the wing demonstrator. Furthermore, the development and integration of the wing concept are discussed for future static and dynamic wind-tunnel experiments at the Open Jet Facility (OJF). The main contribution of this Chapter is threefold. First, an aeroelastic wing demonstrator with distributed control surfaces is designed, analysed, and manufactured. Second, a distributed and decentralised control architecture is proposed and implemented. Third, dynamic closed-loop simulations of the demonstrator were performed in the presence of gusts, verifying the structure and actuator design and highlighting the necessity of distributed control for local load alleviation.

The structure of the Chapter is as follows. The philosophy of SmartX is presented in Sec. IV.1.2, followed by the design methodology in Sec. IV.1.3. The demonstrator manufacturing and integration are presented in Sec. IV.1.4. The results are shown and discussed in Sec. IV.1.5. Finally, conclusions are drawn in Sec. IV.1.6.

IV.1.2. SmartX Philosophy

In the following sections, Sec. IV.1.2.1 and Sec. IV.1.2.2, the aim of the SmartX project and the objectives of the SmartX-Neo are presented and discussed.

IV.1.2.1. Goals of SmartX

The SmartX project aims to demonstrate in-flight performance optimisation of several objectives such as (i) drag optimisation, (ii) load alleviation, (iii) flutter suppression, and (iv) shape control through multidisciplinary integration of control sensing and morphing design. Within the scope of this project, a smart morphing wing was developed: the SmartX-Alpha is capable of continuous active morphing with distributed Translation Induced Camber (TRIC) [8]. The advantage of this design was the capability of local control of the lift distribution along the span through individual adjustment of the camber and twist of each morphing module, allowing the wing to settle into the most optimal lift-to-drag ratio (*shape control*) to *minimise drag* and perform the *load alleviation* tasks [7].

IV.1.2.2. Objectives of SmartX-Neo

With SmartX-Alpha, the first three objectives of the SmartX project were achieved. However, due to limitations of the actuation bandwidth, faster objectives such as flutter suppression were not achievable with morphing alone. The bandwidth limitation arises due to two reasons. Firstly, due to the nature of the TRIC morphing concept, the morphing mechanism of the SmartX-Alpha relies on a *loaded* hinge concept. The skin acts as a hinge between the rigid wing box and the flexible trailing edge (TE), and this requires higher torque from the servo to overcome the internal strain. Higher torque servos generally have to compromise in actuation speed. The current study aims to address this gap by

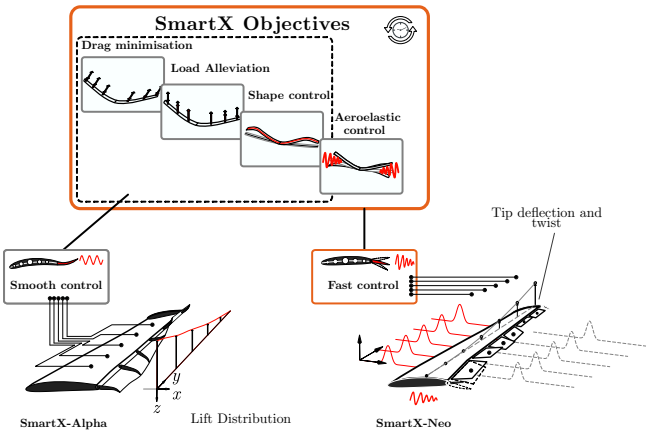


Figure IV.1.1: Comparison of the objectives of the SmartX-Alpha and SmartX-Neo.

investigating the potential of *discrete morphing* with conventionally free hinged flaps. The benefit is significantly lower actuation forces and a simpler actuation mechanism. This yielded the concept of the SmartX-Neo[1] as shown in Fig. IV.1.1.

The SmartX-Neo was developed to investigate the following:

- comparison of *discrete morphing* versus *smooth morphing* in terms of actuation bandwidth and design complexity;
- benefits of conventionally hinged flap versus morphing;
- benefits of over-actuated wing concept for aeroelastic control with advanced control methods;
- influence of the actuation speed on the control objectives.

IV.1.3. Design Methodology

In the following sections, the design methodology of the SmartX-Neo is presented. Section IV.1.3.1 discusses the wing and aircraft planform design. The aeroservoelastic model and the control design are presented in Sec. IV.1.3.2 and Sec. IV.1.3.3. The actuator model and the hypotheses of the numerical experiment are presented in Secs. IV.1.3.4 and IV.1.3.5, respectively.

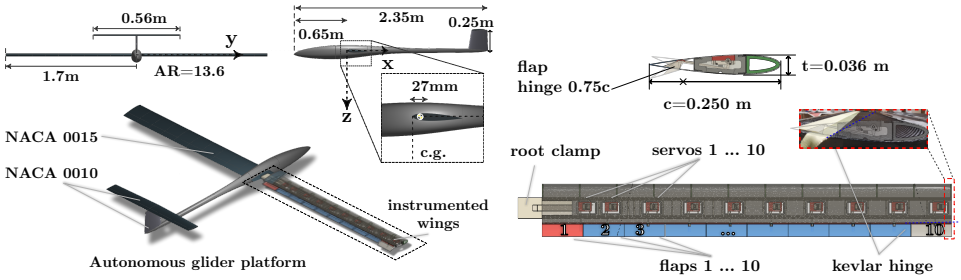
IV.1.3.1. Planform Design and Analysis

The wing design was evaluated using an aerodynamic model, built with XFLR5 [9] and a Finite Element Model (FEM) built with ABAQUS [10], representing the structure of the wing.

[1]The project video can be found at https://www.youtube.com/watch?v=WuxM2vmumkQ

Planform Design

The wing model was designed to investigate the benefits of advanced control methods for over-actuated aeroelastic wings, with the potential of being integrated into an autonomous glider platform. The goal was to build the designed wing to conduct a wind tunnel test in the OJF at the Delft University of Technology, equipped with a gust generator [11]. The wing design is evaluated for the condition of 35 m/s at a cruise angle of 4 degrees. The free stream velocity is chosen with a margin over the wind tunnel's maximum available free stream velocity. The cruise angle of attack is determined through preliminary design and assessment of the suitable flight platform and glider configuration. This is discussed in brief in the following sections.



(a) Aircraft planform design.

(b) Wing planform design.

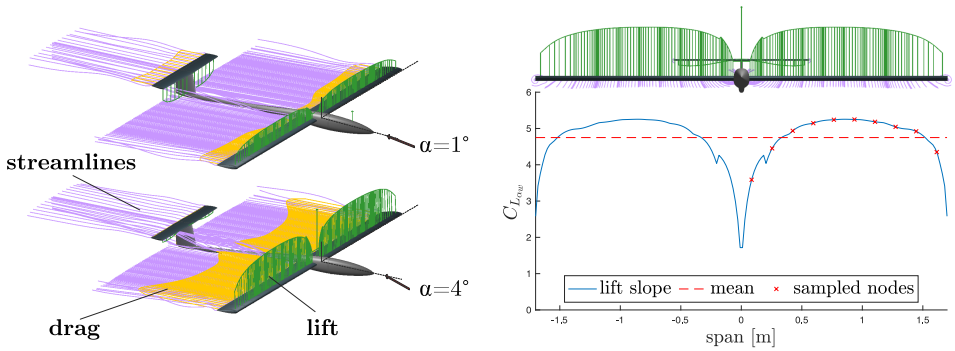
Figure IV.1.2: Overview of the wing planform design.

NACA0015 was selected as the wing profile as a good trade-off between aerodynamic performance and required structural components and instrumentation volume. The span was selected to be 1.7 m considering the manufacturing constraint of the mould. The planform is shown in Fig. IV.1.2.

A preliminary design of the glider platform was conducted in XFLR5. A conventional aircraft configuration was chosen, with the elevator and vertical stabiliser conventionally actuated in a typical glider configuration. A thinner airfoil, NACA0010, was chosen for these wings as there was no requirement for large component volume compared to highly instrumented main wings. The elevator and vertical stabiliser were sized relative to the main wing according to common ratios, such that a balanced design was obtained. The relative placement of the wings and the body was achieved via steady-state stability analysis in XFLR5 at cruise conditions. With a total wing mass of 5.7 kg and payload mass of 1.5 kg, a centre of gravity (COG) x location of 27 mm aft of the main wing leading edge was found, which provided sufficient lateral stability to trim the aircraft in cruise condition at $\alpha = 4°$. The neutral point was found to be 0.305 m aft of the main wing. The remaining parameters are presented in Appendix IV.A1 and Tab. IV.A1.1.

Aerodynamic and Structural Design

To fulfil the requirements of the wind tunnel model and assess the aerodynamic loads expected on the wing structure, as well as the lift generated by the flaps, an aerodynamic model was built using XFLR5. This software is based on the 2D analysis capabilities of the XFOIL code and can implement the Vortex Lattice Method (VLM) and the 3D panel



(a) Aerodynamic analysis for conditions $\alpha = 4°$ and $\alpha = 1°$, at $V_\infty = 35$ m/s.

(b) Lift slope $C_{L_{\alpha_w}}$ distribution main wing used in the aeroservoelastic model, $V_\infty = 35$ m/s, $\alpha = 4°$.

Figure IV.1.3: Aerodynamic analysis and win lift distribution.

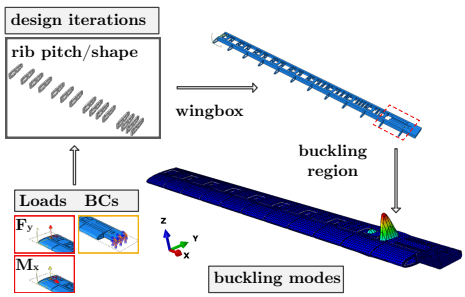
method [9, 12]. The resulting torque on the flap hinge was extracted from the aerodynamic analysis performed in XFLR5, allowing us to evaluate the lift generated by the flap and the wing. The aerodynamic mesh was selected to have 2600 VLM panels and 5225 3D panels, with a 13 (cosine) × 10 (sine) distribution along chord and span.

A type 1 (fixed speed) viscous analysis (Viscosity=1.5e-05 m^2/s) was performed at $V_\infty = 35$ m/s, $\alpha = 4°$ degrees of angle of attack to obtain lift distribution along the span of the wing planform at cruise condition. This is the distribution (blue curve) shown in IV.1.3b, the red curve is the mean distribution used as the wing lift coefficient $C_{L_{\alpha_w}}$. The top figure shows the streamlines (wake) in purple, generated behind the wing and near the wingtips at this condition. Fig. IV.1.3a shows the 3D lift distribution (green), induced drag (yellow) and streamlines (magenta) at $V_\infty = 35$ m/s and two angles of attack, $\alpha = 4°$ and $\alpha = 1°$. The latter angle is selected to show the configuration of the elevator. The figure shows that the elevator is designed with a fixed (4 degrees) negative incidence angle to balance the aircraft at level flight and compensate for the moment generated by the main wing. At cruise (bottom figure), the majority of the lift is generated by the main wing.

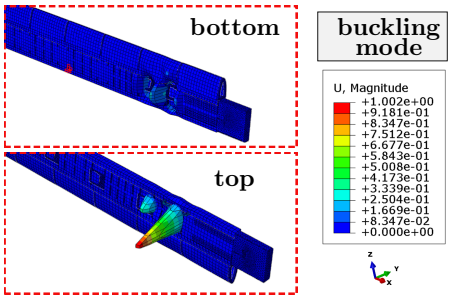
Buckling Analysis

The wing-box structure was constructed to reinforce the structural design. Design iterations were evaluated in terms of buckling resistance. The worst-case condition (i.e. $V_\infty = 50$ m/s, $\alpha = 4°$, $\delta_{\text{flap}} = 25°$) were imposed on the structural FEM model in Abaqus to investigate the buckling behavior. This analysis was necessary due to the flexibility of the structure and the high number of cutouts made in the skin for actuator access bay panels. A limit load of 750 N was established. Figure IV.1.4a shows the setup of the Abaqus model. Two loads were applied, lift load F_y and torsional moment M_x, to make a conservative estimation of the aerodynamic loads induced during the worst-case condition. The boundary conditions (BCs) were imposed to clamp the wing at the root.

Design iteration was made for a suitable rib design, rib pitch, and shape. No buckling occurred below the maximum limits. After initial iterations, the weak point in the



(a) Setup of the Abaqus FEM model for buckling analysis.



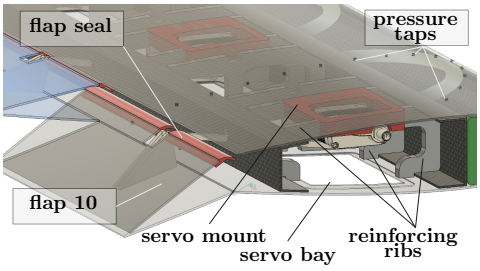
(b) The first buckling mode predicted by the FEM model.

Figure IV.1.4: Buckling model setup and analysis.

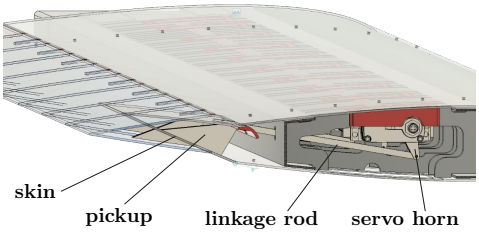
design was found to be near the buckling region, between the first cutout panel for servo 1 (Fig. IV.1.2a) and the wing clamp bonded at the root. Additional reinforcing ribs were added to support the cutout region, as shown in Fig. IV.1.5a. The result of the linear analysis for the first buckling mode with the final wing-box design is shown in Fig. IV.1.4b. The eigenvalue for the first mode was found at -1002.4, well above the limit loads.

Actuator Selection

The actuation loads and the resulting flap moments were evaluated to determine the actuator's torque requirements in XFLR5 and select a suitable actuator. The analysis was conducted at 35 m/s. The angle of attack was maintained at 4°. Figure. IV.1.6 shows a comparison of two servo configurations and the achievable control objectives for SmartX-Alpha versus SmartX-Neo. The right one shows the characteristics of the servo selected for the SmartX-Alpha demonstrated. Here, the actuator torque requirement was evaluated for various morphing conditions of the flaps [5].



(a) Flap close-up view.

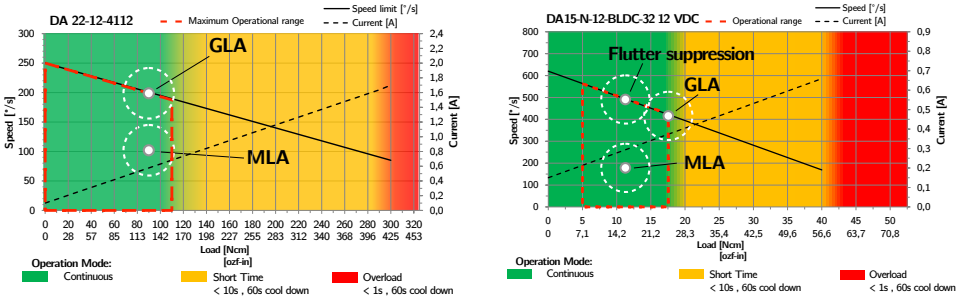


(b) Flap side view.

Figure IV.1.5: Overview of the actuation mechanism.

As opposed to a simple flap design of the SmartX-Neo (Fig. IV.1.5a), the morphing mechanism of the SmartX-Alpha requires relatively high torque servos, which are limited in continuous actuation speed (Fig. IV.1.6a). In turn, the ability to fulfil the control objectives is limited as well. By contrast, the faster servo depicted on the right (Fig. IV.1.6b)

shows higher continuous actuation capability and wider scope of possible control objectives. The white dots indicated in Fig. IV.1.6 are potential operational points described in Fig. IV.1.1. Gust Load Alleviation (GLA) can be maximised at the highest torque and continuous load setting. Manoeuvre Load Alleviation (MLA) is less limiting on actuation speed, as this objective can be achieved in a longer time scale.



(a) Volz DA-22-12-4112 performance parameters [13] and the SmartX-Alpha requirements.

(b) Volz DA-15-N-BLDC performance parameters [13] and the SmartX-Neo requirements.

Figure IV.1.6: Comparison of the actuator continuous torque requirement, SmartX-Alpha versus SmartX-Neo.

The aerodynamic analyses performed for the SmartX-Neo yielded three times faster continuous actuation and six times lower loads on the control surface (10 Nm versus 60-80 Nm) for SmartX-Neo compared to SmartX-Alpha [5]. The actuator bandwidth of the selected servo is presented in Fig. IV.1.6b. The Volz DA-15-N-BLDC servo was selected due to its high continuous load and position feedback capabilities. Figure IV.1.6b shows the performance specification data of the actuator published by the manufacturer [13] and the comparison to the Volz DA-22-12-4112 used for the SmartX-Alpha. The green region indicates the range in which the servo can operate continuously. As seen, the peak torque requirement, indicated with a red-dotted box, falls within the continuous operation range of the servo.

IV.1.3.2. Aeroservoelastic Model

An aeroservoelastic model is developed in Matlab/Simulink to assess actuator requirements concerning the expected dynamic response of the wing demonstrator and to develop a controller capable of fulfilling the objectives of the SmartX-Neo. The model is adapted from [14] and represents a coupled unsteady aeroservoelastic model, trimmed at an air density $\rho_{\text{AIR}} = 1.225$ kg/m^3 and free stream velocity $V_\infty = 35$ m/s. It is composed of ten aerodynamic strips placed at equal distances along the span corresponding to the number of flaps.

Structure

The structure is modelled as a linear Euler-Bernoulli beam. Each actuator (flap) is modelled as a second-order mass-spring-damping system, with a hinge moment control input M_f^{act}. The clamped beam-flap model has four degrees of freedom at each node, represented by the state vector $\mathbf{x}_s = \begin{bmatrix} w & \phi & \theta & \beta \end{bmatrix}^{\mathrm{T}}$. Where, w, ϕ, θ, represent the transverse displace-

ment ($\downarrow +$), bending ($\circlearrowleft +$), and torsion ($\circlearrowleft +$) while β is the flap rotation angle ($\circlearrowleft +$). The dynamics for the clamped beam are given by:

$$\mathbf{M}_s\ddot{\mathbf{x}}_s + \mathbf{C}_s\dot{\mathbf{x}}_s + \mathbf{K}_s\mathbf{x_s} = \begin{bmatrix}\mathbf{F_r} & \mathbf{F_{ext}}\end{bmatrix}^{\mathrm{T}} \tag{IV.1.1}$$

where $\mathbf{M}_s, \mathbf{C}_s, \mathbf{K}_s$ are the structural mass, damping and stiffness matrices, respectively. On the right-hand side, $\mathbf{F_r}$ and $\mathbf{F_{ext}}$ are the wing root reaction forces and the distributed external forces. The wing root reaction forces are, the shear force, the root bending moment, and torsional moment contained in vector $\mathbf{F_r} = \begin{bmatrix}F_w & M_\phi & M_\theta\end{bmatrix}^{\mathrm{T}}$. Structural damping is added proportionally to the stiffness matrix, though $\mathbf{C}_s = k\mathbf{K}_s$, where $\mathbf{C}_s$ is the damping matrix and k a scaling factor.

In Eq. (IV.1.1) beam structural mass and stiffness matrices $\mathbf{M}_s$ and $\mathbf{K}_s$ are augmented to include the effect of the flap, yielding $\mathbf{M}_s^{\text{aug}}$ and $\mathbf{K}_s^{\text{aug}}$, as follows:

$$\mathbf{M}_s^{\text{aug}} = \begin{bmatrix} \begin{bmatrix} & \mathbf{M}_s & \end{bmatrix} & \begin{matrix} S_\beta \\ 0 \\ I_\beta + b(c-a)S_\beta \end{matrix} \\ \begin{matrix} S_\beta & 0 & I_\beta + b(c-a)S_\beta \end{matrix} & I_\beta \end{bmatrix} \tag{IV.1.2}$$

$$\mathbf{K}_s^{\text{aug}} = \begin{bmatrix} \begin{bmatrix} & \mathbf{K}_s & \end{bmatrix} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & K_\beta \end{bmatrix} \tag{IV.1.3}$$

In Eqs. (IV.1.2) and (IV.1.3), the flap angle state β is coupled with the main beam structure through inertia couplings and a rotational spring, serving as actuator stiffness. The measurable outputs are the shear force, the root bending moment, and the node displacements in the heave direction w. In total, ten nodes are movable; the first node, denoted as the 0^{th}, is the reference node at the root (clamped). The remaining nodes are labelled 1-10^{th} and correspond to the centre location of each flap as shown in Fig. IV.1.7.

Aerodynamics

Because of the high aspect ratio of the wing, the two-dimensional strip theory was adopted where the unsteady aerodynamic forces on each strip are represented in a time-domain formulation, equivalent to Theodorsen's frequency-domain model [15]. The time-domain formulation used in this study is the indicial function approximation by Leishman [16].

Referring to Ref. [16], four lag states are introduced for each aerodynamic strip to model the circulatory part of the aerodynamic response. Similar to the structural part, aerodynamic state vector is represented by $\mathbf{x}_a = \begin{bmatrix}w & \phi & \theta & \beta & z_i\end{bmatrix}^{\mathrm{T}}$. Where the latter entry are the lag states. The aerodynamic force vector, $\mathbf{F}_a$, is defined as:

$$\mathbf{F}_a = \mathbf{M}_a\ddot{\mathbf{x}}_a + \mathbf{C}_a\dot{\mathbf{x}}_a + \mathbf{K}_a\mathbf{x}_s + \mathbf{K}_z z_i \tag{IV.1.4}$$

Couplings

The coupling of the structural and the aerodynamic models is described in [14]. The full aeroservoelastic model contains the following states:

$$\mathbf{x}_{ae} = \begin{bmatrix}\dot{\mathbf{x}}_s & \mathbf{x}_s & z_1 & z_2 & z_3 & z_4\end{bmatrix}^{\mathrm{T}} \tag{IV.1.5}$$

IV.1

where $\mathbf{x}_s$ is the structural state vector, representing the nodal degrees of freedom for each of the 11 nodes; z_1, z_2, z_3, z_4 are the aerodynamic lag states.

An overview of the coordinate system, nodes and axis definitions of the aeroelastic system is presented in Fig. IV.1.7. Here O_w represents the right wing frame.

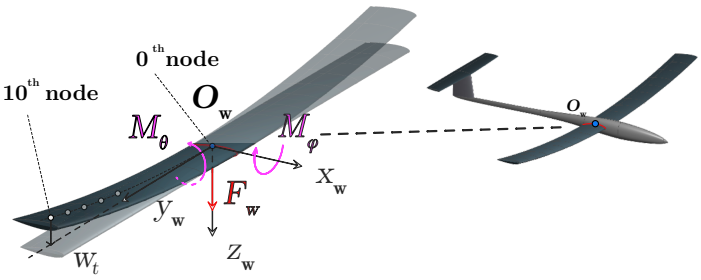


Figure IV.1.7: Reference frames, axis definitions, and degrees of freedom of the aeroelastic system (the right wing).

Gust model

Initial assessment of the dynamic response is performed with a simplified gust model, a "1-cosine" gust profile, assumed to be uniform across the span and represented as an increment in α:

$$\alpha_g(t) = W_g(1 - \cos(\omega_g t)) \tag{IV.1.6}$$

where W_g is the gust magnitude, and ω_g is the gust frequency in radians, defined as $\omega_g = 2\pi f_g$, where f_g is the gust frequency in Hz.

IV.1.3.3. Control design

The aeroservoelastic SmartX-Neo wing is controlled by a Linear-Quadratic Regulator (LQR) controller [17]. This is a linear optimal control method that provides the optimal feedback gain matrix $\mathbf{K}$ to stabilise the system. The aeroservoelastic wing is modelled in a state-space form as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{B}_g\alpha_g \tag{IV.1.7}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \tag{IV.1.8}$$

Where $\mathbf{A}, \mathbf{B}, \mathbf{B}_g, \mathbf{C}, \mathbf{D}$ are the system dynamic matrices, while the gust angle of attack input α_g is defined in Eq. (IV.1.6).

The gain matrix is obtained to minimise the objectives of interest, namely: the wing root shear force F_w, the wing root bending moment M_ϕ, and the wing tip displacement w_t. The state feedback law minimises the quadratic cost function [18]:

$$J = \int_0^\infty \left[\mathbf{x}^\mathrm{T}\mathbf{Q}\mathbf{x} + \mathbf{u}^\mathrm{T}\mathbf{R}\mathbf{u}\right] \mathrm{d}t \tag{IV.1.9}$$

where the weight matrices $\mathbf{Q}$ and $\mathbf{R}$ are positive definite matrices that penalise the cost of deviation of the states from zero and the cost of actuation, respectively. The state feedback

gain matrix that minimises the cost is defined by $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{S}$, where $\mathbf{S}$ is the solution to the Ricatti equation:

$$\mathbf{A}^{\mathrm{T}}\mathbf{S} + \mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{S} + \mathbf{Q} = \mathbf{0} \tag{IV.1.10}$$

The closed-loop system dynamics are:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{B}_g\alpha_g = \underbrace{(\mathbf{A} - \mathbf{B}\mathbf{K})}_{\mathbf{A_{clp}}}\mathbf{x} + \mathbf{B}_g\alpha_g \tag{IV.1.11}$$

In this Chapter, $\mathbf{Q}$ is chosen as $\mathbf{C}^{\mathrm{T}}\mathbf{C}$, while $\mathbf{R}$ is chosen as an identity matrix $\mathbf{I}_n$, with $n = 10$. It is noteworthy that the gust is unknown to the controller. The LQR assumes that full state feedback is available. Suppose full state feedback is not available for a real-life system, then the Linear Quadratic Gaussian (LQG) control law can be designed, which contains a Kalman filter state estimations using a combination of sensory measurements (e.g., cameras, gyros).

IV.1.3.4. Actuator dynamics

One of the objectives of this study is to investigate how design changes in actuation can affect the performance of control objectives (Fig. IV.1.1). In particular, the study aims to assess the role of faster actuation for achieving these control objectives. It is to be expected that the servo bandwidth influences the performance of gust load alleviation. Therefore, a parametric model of actuator dynamics is constructed to investigate its influence. Actuator dynamics can be modelled by a second-order system, which is analogous to a mass-spring-damping system. To parametrically adjust the damping and stiffness parameters, a parameter k is chosen, with the following relationship:

$$\mathbf{K}_\beta = \mathbf{K}_f k^2, \quad \mathbf{C}_\beta = \mathbf{C}_f k \tag{IV.1.12}$$

where $\mathbf{K}_\beta$ and $\mathbf{C}_\beta$ are the stiffness and damping matrices in the augmented structural model corresponding to the entries of the flap. The matrices $\mathbf{K}_f$ and $\mathbf{C}_f$ are the original actuator stiffness and damping matrices. Equation (IV.1.12) ensures that the natural frequency of the actuator dynamics is scaled proportionally while the damping ratio is kept invariant.

IV.1.3.5. Hypotheses

The analysis performed in this study shall be limited to Gust Load Alleviation (GLA) and the influence of actuator dynamics. Considering these constraints and the assumptions made in the aeroservoelastic model and the actuator model, two hypotheses are formulated:

1. The first hypothesis is that higher actuator bandwidth will be more effective for GLA;
2. The second hypothesis is that a higher actuator bandwidth shall allow more effective GLA at higher gust frequencies.

The last hypothesis means that we expect that faster actuators will allow the controller to respond faster to more high-frequency disturbances encountered by the system. In Sec. IV.1.5 simulations are set up to investigate the hypotheses.

IV.1

IV.1.4. Demonstrator Manufacturing and Integration

The design analysis discussed in Sec. IV.1.3 was used to support the manufacturing of a composite wing demonstrator, SmartX-Neo, with a wing-box structure and an integrated actuation mechanism.

IV.1.4.1. Manufacturing Process

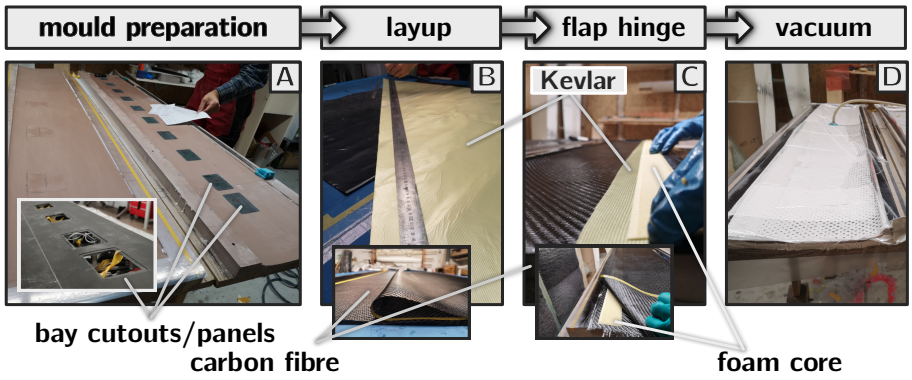


Figure IV.1.8: Manufacturing process.

The composite wing design was constructed in four parts (i) top skin, (ii) bottom skin, (iii) wing box structure, and (iv) distributed flap modules. A mould consisting of top and bottom halves was manufactured out of Polyurethane based SikaBlock with a density of 650 kg/m^3 [19]. The composite wing skin consisted of three layers of 160 g/m^2 carbon fibre and additional fibreglass of 40 g/m^2 for a smooth surface finish. The skin was manufactured and cured with the hand layup technique in the top and bottom vacuum bagged moulds as shown in Fig. IV.1.8. The spars were cured in a separate mould and made of 2 layers of 160 g/m^2 carbon fibre and assembled into the main wing structure.

Figure IV.1.8 shows the first phase of the manufacturing process in sub-figures A-D. This process is initiated by the preparation of the mould (Fig. IV.1.8-A), where the pre-manufactured servo panel covers are arranged at the location of the cutouts. The panels are placed underneath the initial layer such that the actual panel cutouts are formed to the specifications and the surface is smooth. Ten cutouts were made to facilitate actuator maintenance and assembly, corresponding to 10 flaps per wing. The flaps are numbered as 1-10, with the 10 th flap corresponding to the outermost flap of module 10 and the 1 th flap the one closest to the fuselage, as indicated by the red colour in Fig. IV.1.2a. Process A is followed by hand layup (Fig. IV.1.8-B), where the flap hinge material is stacked between the carbon fibre layers.

The hinge for the flap mechanism was based on foam-reinforced Kevlar material. A single strip of 110 g/m^2 Kevlar was added between carbon fibre layers 2 and 3 (most inner layer). Additionally, a Herex foam strip was added for additional stiffness, as depicted in Fig. IV.1.8-C. The process is completed by vacuum bagging and curing (Fig. IV.1.8-D).

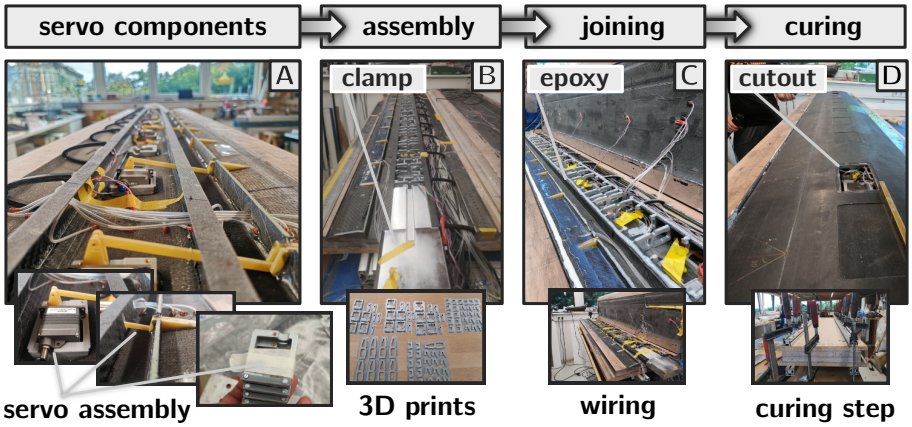


Figure IV.1.9: Assembly integration process.

IV.1.4.2. Integration Process

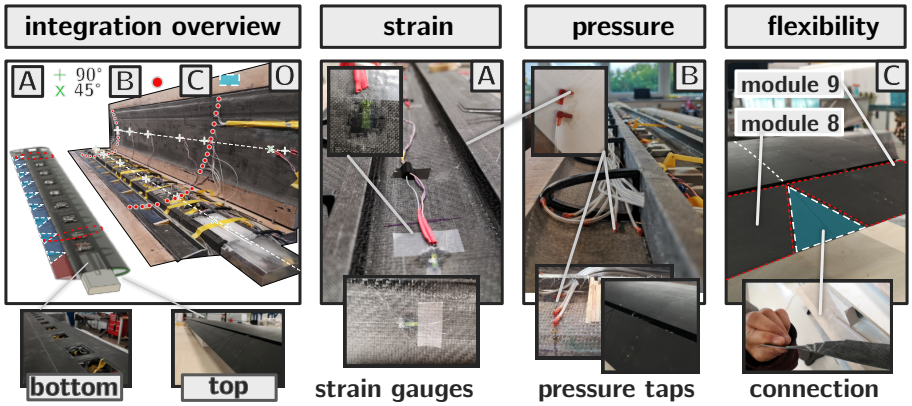


Figure IV.1.10: Module and sensor integration.

The assembly and integration process is depicted in sub-figures A-D of Fig. IV.1.9. The actuation mechanism was designed such that it was fully integrated inside the wing and could be assembled after the joining of top and bottom parts (Fig. IV.1.9-A). The parts were 3D printed from polylactic acid (PLA) and used in the assembly process as shown in Fig. IV.1.9-B. A 3D-printed base was designed to house the actuators and bonded to the top skin, shown in the bottom part of sub-Fig. IV.1.9-A. To resist buckling, a supporting rib structure was added near the cutouts and bonded to the skin.

The assembly process is shown in Fig. IV.1.9-C. Here, two wing halves were joined after the wiring and all assembly components were in place. To bond the two wing halves, epoxy was deposited on the contact surfaces between the wing-box. Epoxy-infused foam

cores were arranged along the wing seams to ensure a stronger bond. The final step in the integration was the curing and demoulding (Fig. IV.1.9-D).

The sensor and module integration process is depicted in sub-figures A-D of Fig. IV.1.10. An overview of the various sensors and their arrangement in the wing is illustrated in Fig. IV.1.10-A. Strain gauges (Fig. IV.1.10-B) were installed on the top and the bottom in 90° arrangement (shown as a plus) and 45° arrangement to measure the twist near the root (shown as a cross). In addition to the strain gauges, two fibre optics sensors were installed along the span on the top and bottom sides. Furthermore, pressure taps were installed (Fig. IV.1.10-C) to allow characterisation of the pressure distribution in the wind tunnel test. These were arranged in an array of 22-28 taps per airfoil cross-section at approximately 15° angle with respect to the free stream velocity. This was done such that the interference in pressure measurements along the chordwise taps was reduced. The tubing was guided through the wing root and D-box area at the root of the wing.

Flexible feather-like patches were integrated between the modules for improved smoothness and aerodynamic properties of the flaps. Figure IV.1.10-D shows the triangular segment between the outer flap (9) and the adjacent flap (8). The segments were made of the same material as the skin, supported by elastomeric filler material. This filler material (Elastosil E41) was deposited during the integration phase from the inner side of the flap. Triangular cutouts were then made to facilitate flexibility.

IV.1.4.3. System Control Architecture

An overview of the control architecture of the glider platform is visualised in Fig. IV.1.10. In this figure, the blocks AFCS and SFC represent the Automatic Flight Control System (AFCS), responsible for the controller, and the Sensor Fusion Computer (SFC), responsible for processing the multitude of sensor data (camera, strain gauge, pressure sensors, etc.), respectively. A distributed data-sharing architecture is developed based on the decentralised communication principle to facilitate smooth and adaptable integration of over-actuated wing systems and the multitude of sensors in a real-time operation. This principle was investigated in sensor-based distributed control of the SmartX-Alpha demonstrated in a wind tunnel experiment in OJF [7]. Based on shared memory structure, the principle allows parallel integration of hardware and software components in various programming languages (Python, Matlab, Simulink, C++, .NET, etc.) and various communication protocols (RS485, Ethernet, ModBus, etc.).

The architecture software is developed in C++ with the real-time D-SIM framework, connecting several PC nodes over a local Ethernet network [20], and enabling synchronisation as depicted in Fig. IV.1.10. Here, the blue line is the data-sharing bus that facilitates the integration of modular hardware and software components.

The benefits of the proposed approach include: (i) the control functions have the flexibility to choose their inherent sampling rates; (ii) each sensor can be sampled at the optimal sampling rate of the sensor with its dedicated hardware (ADC converter etc.); (iii) controller tuning with hardware-in-the-loop (HIL) becomes very flexible; (iv) ease of integration and up-scaling of the system with additional sensors; (v) allows implementation of robust sensor fusion algorithms.

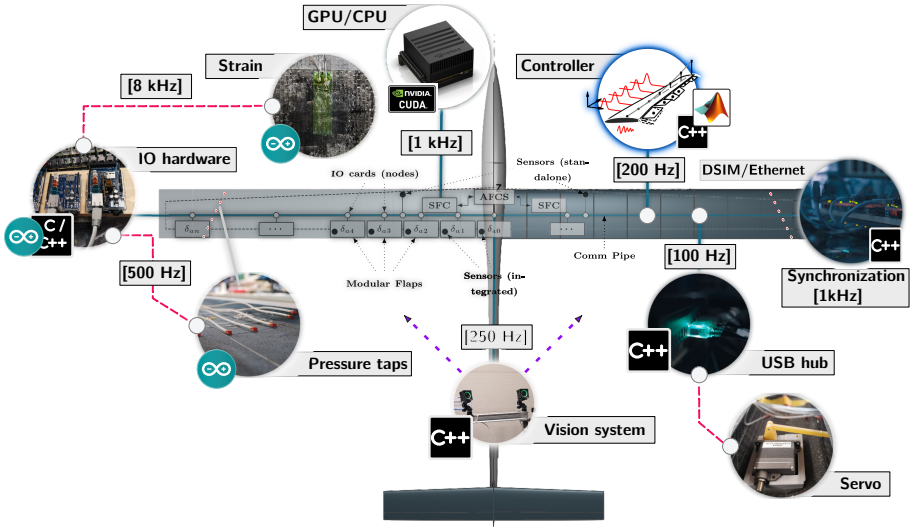


Figure IV.1.11: Overview of the distributed and decentralised control architecture.

IV.1.5. Results and Discussions

In this section, the analysis of the design and the results of the simulation experiment are discussed. First, the effect of the actuator dynamics on the performance of the baseline LQR controller is discussed. Then the analysis of the wing planform design and aircraft characteristics are discussed.

IV.1.5.1. Gust Load Alleviation

A state-space aeroelastic dynamic model was established to analyze the gust response and control effectiveness of the wing. The aeroelastic model, as described in section IV.1.3.2 was subjected to discrete gust signals. The actuator dynamics implemented in the aeroservoelastic model were modified through the scaling parameter k governing the actuation design and dynamics (Sec. IV.1.3.4).

Simulation Setup

Table IV.1.1: Simulation and design configurations.

	Design	Simulation			
	k [-]	f_g [Hz]	W_g [rad]	dt [s]	t_{sim} [s]
input	1.00	1.00	$2/V_\infty$	0.001	1.20
	0.75	2.00	$2/V_\infty$	0.001	1.00
	0.50	3.00	$2/V_\infty$	0.001	1.00

Three scaling values were chosen; additionally, three different gust frequencies were evaluated to assess the effectiveness of the controller, resulting in a total of 9 different simulation conditions. In each simulation, the control gain matrix was kept constant as described in Sec. IV.1.3.3 and an actuation limit of 25 degrees of flap deflection was considered. The simulation was evaluated at $\rho_{\text{air}} = 1.225$ kg/m^3 and $V_\infty = 35$ m/s. For slower gust (f_g = 1 Hz), a longer simulation time was used to ensure the entire gust onset was captured. The varied input conditions are presented in Table IV.1.1. Each row of the design parameter k was varied with the entries of the simulation parameters $(f_g, W_g, \mathrm{d}t, t_{\text{sim}})$.

The first simulation run would thus consist of the first row of Table IV.1.1, and the second would be composed of the parameter sets: $k = 0.75,\ f_g = 1$ Hz, $W_g = 2/35$ rad, $\mathrm{d}t = 0.001$ s, $t_{\text{sim}} = 1.20$ s. The parameters of the aeroservoelastic model, which were kept invariant, are presented in Table IV.1.2.

Table IV.1.2: Parameters of aeroservoelastic model.

Parameter	Symbol	Value	Unit
Half span	L_w	1.70	[m]
Chord	$2b$	0.25	[m]
Shear centre location	a	0.00	[m]
Wing lift coefficient	$C_{L_{\alpha_w}}$	4.75	[-]
Mass per unit length	m	0.75	[kg/m]
Flap mass per unit length	m_f	0.25	[kg/m]
Moment of inertia	I_θ	0.24	[kgm^2]
Bending stiffness	EI	$1/2 \cdot 10^3$	[Nm^2/m]
Torsional stiffness	GJ	10^5	[Nm/m]

In Eq. (IV.1.12), $\mathbf{K}_f = 100$ and $\mathbf{C}_f = 10^{-4}$. To evaluate the performance of the closed-loop system, two metrics are used: the maximum percentage difference of the peak f_{max} and the area difference f_{area} between closed- and open-loop responses. For example, for the shear force F_w, the performance metric is defined as:

$$f_{\text{max}}(\Delta F_w) = \left(1 - \frac{\max(|F_w|)_{\text{closed}}}{\max(|F_w|)_{\text{open}}}\right) \cdot 100\% \tag{IV.1.13}$$

The area difference defined for the shear force F_w is:

$$f_{\text{area}}(\Delta F_w) = \left(1 - \frac{\text{area}(|F_w|)_{\text{closed}}}{\text{area}(|F_w|)_{\text{open}}}\right) \cdot 100\% \tag{IV.1.14}$$

In Tables IV.1.3 and IV.1.4 the operator $f(\cdot)$ is used to describe the operation needed to obtain the aforementioned metrics.

Effects of Gust Frequency

The effectiveness of the controller for gust load reduction was evaluated for various gust inputs with parameters described in Table IV.1.1. The result of simulations for the metrics peak magnitude and peak area are presented Tables IV.1.3 and IV.1.4, respectively. Herein,

the values indicated are the percentage reduction of the specified metric as compared to the open-loop. Three gust load reduction objectives were considered, the shear force F_w, root bending moment M_ϕ, and the tip displacement w_t. The rows in the table indicate variations in the scaling parameter k.

Table IV.1.3: Gust load reduction comparison for metric peak magnitude for varying gust frequency and scaling.

Variables	f_g = 1.00 [Hz]			f_g = 3.00 [Hz]			f_g = 5.00 [Hz]		
	$f(\Delta F_{w_*})$	$f(\Delta M_{\phi_*})$	$f(\Delta w_{t_*})$	$f(\Delta F_{w_*})$	$f(\Delta M_{\phi_*})$	$f(\Delta w_{t_*})$	$f(\Delta F_{w_*})$	$f(\Delta M_{\phi_*})$	$f(\Delta w_{t_*})$
k=1.00	71.26	78.25	65.00	69.67	76.26	63.25	67.39	73.20	60.71
k=0.75	77.15	82.88	70.22	75.71	81.30	68.58	73.50	78.54	65.99
k=0.50	82.30	83.84	69.24	81.14	82.46	67.83	79.23	79.85	65.06
k=0.75 [$\Delta_{k=1}$]	+5.89	+4.63	+5.22	+6.04	+5.03	+5.33	+6.11	+5.34	+5.29
k=0.50 [$\Delta_{k=1}$]	+11.04	+5.59	+4.24	+11.47	+6.19	+4.58	+11.83	+6.65	+4.35

Looking at the first row, corresponding to the nominal actuator dynamics ($k = 1$), it is observed that significant reductions (65-85 %) are achieved for all metrics and objectives with the closed-loop control. Furthermore, it is observed that the reduction is consistently less effective for increasing gust frequencies (e.g., 71.78 %, 69.67 %, 67.39 % for shear force) for all objectives for the metric peak magnitude. This is consistent with the expectation that the higher gust frequency will induce a sharper disturbance onset from the wing in the open-loop, requiring faster effort by the controller (faster response). The time responses to the gust onset are plotted for the three objectives and the highest and lowest gust frequency in Figs. IV.1.12 and IV.1.13, respectively. The gust input also follows a similar shape to the open-loop response, indicated by the blue dashed curve. The columns in these plots correspond to shear force, bending moment, and tip deflection, respectively. Here, it is observed that the peak magnitude for the open-loop is smaller for slower gusts. Besides, the response is also narrower. The effectiveness of GLA is determined by the ability of the controller to flatten the onset peak and reduce the area underneath the curve. The closed-loop responses at varying scales clearly support the effectiveness (e.g., $\approx$ 180 N versus $\approx$ 40 N for the shear force at 1 Hz).

Flap Deflections

Given that the amount of disturbance the controller can resist is constrained by the bandwidth of the actuator –in this model governed by a second-order system actuator dynamics– the effectiveness decreases as gust frequency increases.

The flap allocation corresponding to the gust onset and responses in Figs. IV.1.12 and IV.1.13 are plotted in Figs. IV.1.15a and IV.1.15b, where the flap location corresponding to the flap distribution along the span of the wing, are differentiated by colour. The shape of the line differentiates the varying scaling, dash-dotted corresponding to the fastest actuator. The distributions observed for one and 5 Hz gusts, respectively, show significantly narrower flap onset and more actuator input to counter the gust (e.g., $\approx 15^\circ$ versus $\approx 25^\circ$) for inboard flaps. Furthermore, observing the differences between the nominal and fastest actuator (dash-dotted line), the flap angles are twice as high at k=0.50 (e.g., $\approx 10^\circ$ versus $\approx 20^\circ$ for flap four at 1 Hz).

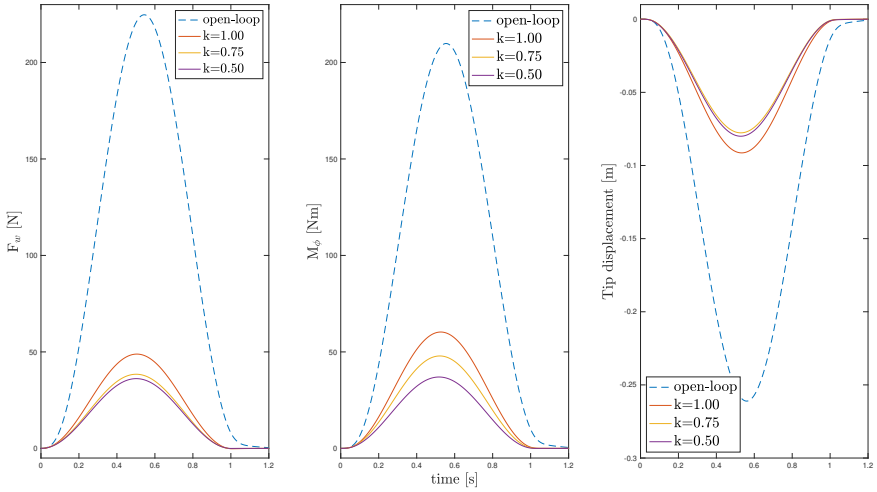


Figure IV.1.12: Open and closed-loop wing response comparisons at gust frequency 1 Hz.

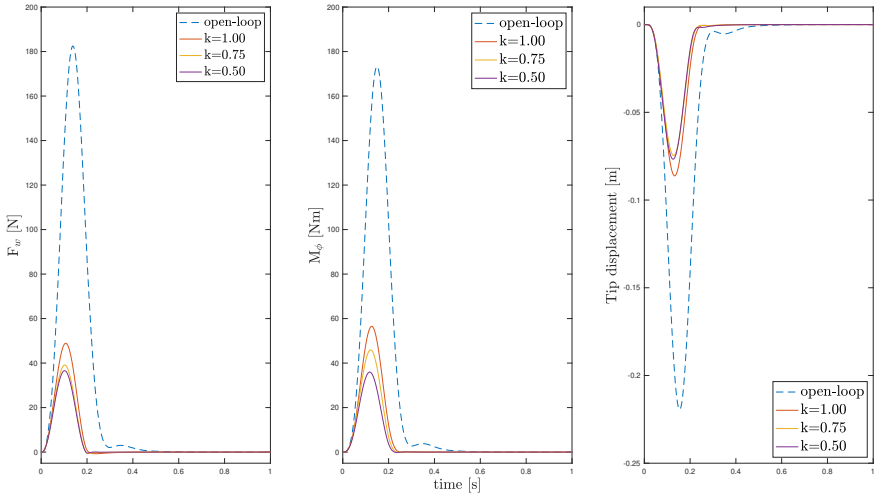


Figure IV.1.13: Open and closed-loop wing response comparisons at gust frequency 5 Hz.

Effect of Faster Actuator Dynamics

Consequently, as the bandwidth of the actuator is increased, the controller can react to the gust disturbance, and therefore the effectiveness of the GLA should increase. Higher reduction percentages observed for lower scaling parameters, corresponding to rows in the mentioned tables, Tables IV.1.3 and IV.1.4, confirm this hypothesis. The fourth and fifth rows in these tables represent the percentage difference of reduction compared to slower nominal dynamics; +, indicating an improved delta. Here we see significant im-

provements in the GLA for all objectives. In particular, we see significant improvements in shear force and bending moment reduction (e.g., 5.89 %, 11.04 % for the shear force at 1 Hz) for increasing k.

A second critical observation can be made from the comparison of the deltas for each objective across varying frequencies. It is clear that the faster actuator positively impacts the controller effectiveness (e.g., 11.04 %, 11.47 %,11.83 % for the shear force at k=0.50 [$\Delta_{k=1}$]) for faster gust onsets, confirming the second hypothesis, namely, that disturbance rejection for higher actuator bandwidth will be more apparent at higher gust frequencies. Differences are also observed across the objectives. Bending moment and tip displacement deltas follow the same trend, meaning better reduction for faster actuator but lower magnitude. The only exception to this lower positive delta reduction for k=0.75 compared to k=0.5 (e.g., 5.89 %, 11.04 % at 1 Hz) for the tip displacement. However, the faster actuator is still beneficial at higher gust frequencies even for this objective (e.g., +5.22 %,5.33 %,5.29 % k=0.50 [$\Delta_{k=1}$]).

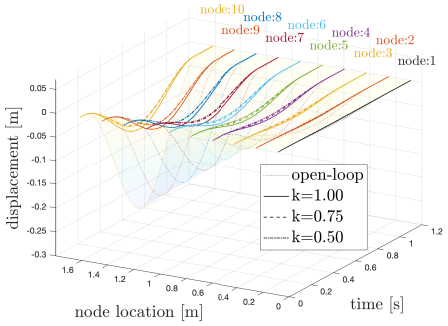
A possible explanation for the differences observed between the objectives is the time scale that characterises their responses. The shear forces directly relate to instantaneous acceleration triggered by a gust event, being the faster the one and thus more apparent to reduce. Similar observations are made from the time responses for closed-loop at different scaling parameters (Figs. IV.1.12 and IV.1.13). Here, the curves corresponding to decreasing scaling parameter k are indicated by red, yellow, and magenta curves. In all cases, the magenta curve (the fastest actuator) is the flattest curve except for the tip displacement.

Differences in Metrics and Distributions

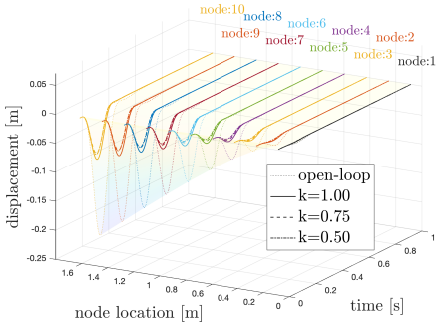
Table IV.1.4: Gust load reduction comparison for metric area for varying gust frequency and scaling.

Variables	f_g = 1.00 [Hz]			f_g = 3.00 [Hz]			f_g = 5.00 [Hz]		
	$f(\Delta F_{w_*})$	$f(\Delta M_{\phi_*})$	$f(\Delta w_{t_*})$	$f(\Delta F_{w_*})$	$f(\Delta M_{\phi_*})$	$f(\Delta w_{t_*})$	$f(\Delta F_{w_*})$	$f(\Delta M_{\phi_*})$	$f(\Delta w_{t_*})$
k=1.00 [-]	71.68	78.70	65.48	71.69	78.70	65.50	71.69	78.70	65.50
k =0.75 [-]	77.51	83.20	70.65	77.51	83.21	70.66	77.51	83.21	70.66
k=0.50 [-]	82.58	84.11	69.59	82.58	84.12	69.60	82.58	84.12	69.60
k=0.75 [$\Delta_{k=1}$]	+5.83	+4.51	+5.17	+5.82	+4.51	+5.17	+5.82	+4.51	+5.16
k=0.50 [$\Delta_{k=1}$]	+10.90	+5.42	+4.11	+10.89	+5.42	+4.11	+10.89	+5.42	+4.11

In Table IV.1.4 the result is shown for the metric area. Whereas the max peak metric shown in Table IV.1.3 is related to onset maxima and minima, the area metric is related to the shape of the response (area under the curve comparison) as shown in responses in Figs. IV.1.12 and IV.1.13. Observing the shape of the curve between gust frequencies 1 and 5 Hz, it is seen that the gust onset in open-loop versus closed-loop scales consistently. This can explain why the area metric seems invariant to the gust frequency (e.g., the variations of the percentage differences are similar across gust frequencies). Observing the deltas in rows 4 and 5, we see a similar trend between the objectives (higher shear force reduction) and higher effectiveness for a faster actuator. It can also be concluded from these observations that, in general, the gust onset reduction mechanism of the LQR is aimed at reducing the area underneath the peak compared to the open-loop. The LQR controller achieves this consistently for various gust frequencies. It should be noted that the LQR controller was kept invariant in these simulation scenarios.

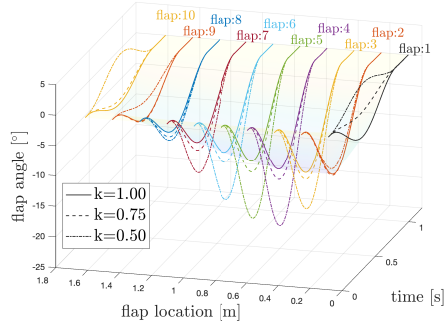


(a) Wing node displacements 1 Hz.

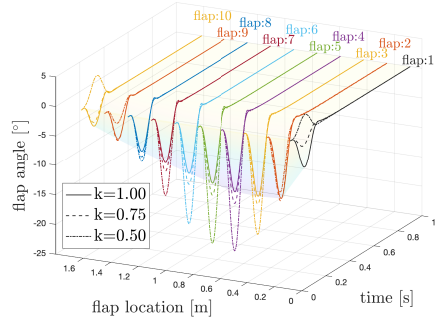


(b) Wing node displacements 5 Hz.

Figure IV.1.14: Spanwise wing node displacements comparison for two frequencies and varying scaling parameter.



(a) Flap deflections 1 Hz.



(b) Flap deflections 5 Hz.

Figure IV.1.15: Spanwise flap deflections comparison for two frequencies and varying scaling parameter.

IV.1.5.2. Initial Design Evaluation

Preliminary evaluation of the wing and planform design was performed based on the numerical simulations described in the previous section. The control allocation, particularly the maximum flap angles at the peak of the onset dictated by the GLA controller for gusts of 1 Hz and 5 Hz, was evaluated with the aircraft planform. These allocations are shown in Figures IV.1.15a and IV.1.15b. The flap deflections at transient peaks (maximum flap deflection) for $k = 0.50$ are shown in Table IV.1.5. For 1 Hz and 5 Hz, these are at ≈ 0.5 and ≈ 0.125, respectively.

Table IV.1.5: Peak flap deflections ([deg]) for 1 Hz and 5 Hz gust frequencies at $k = 0.50$ and $V_\infty = 35$ m/s.

	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_10
1.00 Hz	2.9904	-10.976	-18.144	-21.322	-20.47	-17.76	-13.624	-8.4668	-2.4377	4.5271
5.00 Hz	2.9722	-10.556	-18.451	-21.843	-21.152	-18.448	-14.248	-8.9932	-2.8749	4.1325

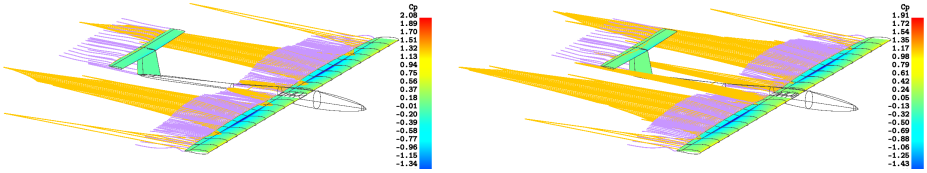
The analysis of the resulting actuator allocations was performed in XFLR5 at $V_\infty = 35$ m/s at cruise conditions. In particular, an initial assessment of the pressure distribution, induced drag, and the wake behaviour behind the wing were studied. Figures IV.1.16a and IV.1.16b show the wake characteristics of the corresponding gust frequencies. Figures IV.1.17a and IV.1.17b show the pressure distributions of the corresponding frequencies.

(a) Wake at 1 Hz.

(b) Wake at 5 Hz.

Figure IV.1.16: Wake characteristics at $V_\infty = 35$ m/s corresponding to the flap angles in Table IV.1.5.

As can be seen from Table IV.1.5, the deflections are comparable but relatively higher for the 5 Hz frequencies. The corresponding lift distributions observed from the XFLR5 analysis are also sharper. In both cases, a low-pressure region is generated by upward deflections of the inboard flaps, which consequently reduces the lift to counter the incoming gust. Due to more extensive flap settings, more considerable induced drag (yellow) and wake (purple) are generated behind the wing at 5 Hz. It must be noted that the aerodynamic solution found for these cases is a steady-state solution and does not include the effect of transient allocation. However, these initial analysis results suggested that the allocation cannot be carried out without an additional drag penalty. Therefore, additional analysis is recommended to study the drag penalty associated with the flap deflections and possibly include this as an additional objective for the LQR controller.



(a) Pressure distributions at 1 Hz.

(b) Pressure distributions at 5 Hz.

Figure IV.1.17: Pressure distributions and streamlines at $V_\infty = 35$ m/s corresponding to the flap angles in Table IV.1.5.

IV.1.6. Conclusions and Recommendations

A distributed over-actuated aeroelastic wing demonstrator was developed within the framework of the Smart-X project, aiming to demonstrate in-flight performance optimisation and multi-objective control with over-actuated wing designs. Aerodynamic and structural analyses were performed to determine actuator torque requirements and actuation mechanism design. The effect of actuator design was studied through a series of gust simulations of closed-loop control of a parametric aeroservoelastic model for gusts with various frequencies. Actuator dynamics were implemented in the system through scaling of the stiffness and damping of a second-order system. With the nominal LQR controller, reductions of peak gust load up to 78 % were achieved compared to the open-loop case.

It was observed that lower scaling corresponding to faster actuation provides significant improvements of up to 11 % over the nominal actuator configuration, yielding reductions of gust loads up to 84 %. Furthermore, it was observed that the effectiveness of faster actuators improves for higher frequency gusts. This confirms the potential of SmartX-Neo to deal with faster control objectives more effectively.

It must be noted that the result presented here consider a purely numerical scenario with full state feedback of the system. In a more realistic case, the noise and bandwidth limitations of the sensors will impact the effectiveness of GLA. It is recommended to include and study these effects in further research. Furthermore, it is recommended to evaluate the prioritisation of objectives on the effectiveness of gust load alleviation.

Finally, the manufacturing and integration of the wing demonstrator were discussed in preparation for future static and dynamic wind-tunnel tests at the OJF wind tunnel.

References

[1] T. Mkhoyan, X. Wang, and R. De Breuker, "Aeroelastic Wing Demonstrator with a Distributed and Decentralized Control Architecture," *AIAA SciTech Forum 2022; Submitted to AIAA Journal*, p. 1551, 2022.

[2] D. Lentink, U. K. Müller, E. J. Stamhuis, R. De Kat, W. Van Gestel, L. L. M. Veldhuis, P. Henningsson, A. Hedenström, J. J. Videler, and J. L. Van Leeuwen, "How swifts control their glide performance with morphing wings," *Nature*, vol. 446, no. 7139, pp. 1082–1085, 2007.

[3] P. Henningsson, A. Hedenström, and R. J. Bomphrey, "Efficiency of Lift Production in Flapping and Gliding Flight of Swifts," *PLoS ONE*, vol. 9, p. e90170, Feb. 2014.

[4] H. R. Jex and F. E. C. Culick, "Flight Control Dynamics of the 1903 Wright Flyer," in *12th Atmospheric Flight Mechanics Conference*, (Reston, Virigina), pp. 534–548, American Institute of Aeronautics and Astronautics, Aug. 1985.

[5] T. Mkhoyan, N. R. Thakrar, R. De Breuker, and J. Sodja, "Design and Development of a Seamless Smart Morphing Wing Using Distributed Trailing Edge Camber Morphing for Active Control," in *AIAA Scitech 2021 Forum*, p. 0477, 2021.

[6] T. Mkhoyan, N. R. Thakrar, R. De Breuker, and J. Sodja, "Design of a Smart Morphing Wing Using Integrated and Distributed Trailing Edge Camber Morphing," in *ASME 2020 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, pp. –, American Society of Mechanical Engineers, sep 2020.

[7] X. Wang, T. Mkhoyan, I. Mkhoyan, and R. De Breuker, "Seamless Active Morphing Wing Simultaneous Gust and Maneuver Load Alleviation," *Journal of Guidance, Control, and Dynamics*, vol. 44, pp. 1649–1662, sep 2021.

[8] N. Werter, J. Sodja, G. Spirlet, and R. De Breuker, "Design and Experiments of a Warp Induced Camber and Twist Morphing Leading and Trailing Edge Device," in *24th AIAA/AHS Adaptive Structures Conference*, (San Diego, California, USA), American Institute of Aeronautics and Astronautics, Jan. 2016.

[9] A. Deperrois, *XFLR5 Theoretical overview*, 2021.

[10] Dassault Systèmes, *ABAQUS/CAE Theory Guide, Version 6.19*. Dassault Systèmes, 2019.

[11] K. Jongkind, A. Falkmann, and H. van der Veer, "Open Jet Facility," 2020.

[12] M. Drela, "XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils," in *Low Reynolds Number Aerodynamics*, vol. 54, pp. 1–12, Berlin, Heidelberg: Springer Berlin Heidelberg, 1989. Series Title: Lecture Notes in Engineering.

[13] Volz, "Volz DA 22 Actuator Technical Specification," tech. rep., Volz Servos GmbH, 2013.

[14] R. De Breuker, S. Binder, and A. Wildschek, "Combined Active and Passive Loads Alleviation through Aeroelastic Tailoring and Control Surface/Control System Optimization," in *2018 AIAA Aerospace Sciences Meeting*, (Kissimmee, Florida), American Institute of Aeronautics and Astronautics, Jan. 2018.

[15] T. Theodorsen, "General Theory of Aerodynamic Instability and the Mechanism of Flutter," tech. rep., NACA Technical Report No. 496 (NACA-TR-496), 1935.

[16] J. G. Leishman, "Subsonic unsteady aerodynamics caused by gusts using the indicial method," *Journal of Aircraft*, vol. 33, pp. 869–879, Sept. 1996.

[17] X. Wang, T. Mkhoyan, and R. De Breuker, "Nonlinear Incremental Control for Flexible Aircraft Trajectory Tracking and Load Alleviation," *Journal of Guidance, Control, and Dynamics*, vol. 1, pp. 1–19, aug 2021.

[18] Y. Ferrier, N. T. Nguyen, E. Ting, D. Chaparro, X. Wang, C. C. de Visser, and Q. P. Chu, "Active Gust Load Alleviation of High-Aspect Ratio Flexible Wing Aircraft," in *2018 AIAA Guidance, Navigation, and Control Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, Jan. 2018.

[19] Sika, "Model and Mold Manufacturing," 2021.

[20] multiSIM, "multiSIM: Distributed simulation D-SIM," 2021.

IV.2

Conclusions and Recommendations

Any product that needs a manual to work is broken.

Elon Musk

Following the findings presented in the previous Parts and Chapters, the conclusions of this dissertation are organised into three Parts. Firstly, a reflection is made on the research question posed in Chapter 1 and the objectives achieved in this dissertation. Secondly, following the representation of an active morphing system in Fig. 1.1 in terms of primary building blocks, each block's main findings corresponding to the dissertation's Parts are summarised. Finally, a prospective outlook of future research in active morphing systems is outlined, and recommendations are made.

Significant advancements have been made in morphing distributed aeroservoelastic systems, sensors, and control methodologies in the past decade. However, the literature review highlighted the need for morphing and aeroelastic systems that are actively controlled to maximise fuel efficiency and reduce structural weight. Furthermore, a holistic approach was emphasised where novel sensing, design and manufacturing, control and optimisation methodologies are considered and integrated simultaneously into an active morphing system. Therefore the following research goal was established:

Research Goal

Develop the smart morphing framework for real-time in-flight performance optimisation, through multidisciplinary parallel integration of sensor-based control laws and the smart distributed sensing and actuation systems.

It was established that simultaneous integration of the *building blocks* of the active morphing wing was crucial to realising this goal. Therefore, the following research question was formulated:

Research Question

How can multidisciplinary integration of sensor-based control laws, model-free sensing methods, and actuation mechanisms be used for real-time, in-flight, multi-objective optimisation framework of actively morphing wings?

IV.2.1. Conclusions

In the following sections, the main findings of the previous Chapters are presented, followed by a future outlook.

IV.2.1.1. Part I: Smart Sensing: Visual Tracking

In Part I, a non-invasive sensing approach was investigated and proposed to estimate the states of morphing and flexible wings in real-time, suitable for control feedback. Novel algorithms were proposed using a holistic approach from the domains of state estimation, machine learning, computer vision and deep learning.

Chapter I.1: Adaptive Real-Time Clustering for Visual Tracking

This Chapter developed a non-invasive vision-based image tracking pipeline using a robust machine learning approach to automatically detect and label visual markers from an image stream and integrate state estimation routing into the control feedback loop. An emphasis was put on robustness and the ability of the algorithm to deal with image noise. A novel approach was proposed using an inverse formulation of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN), DBSCAN^{-1}. Figure IV.2.1 illustrated the focus of Chapter I.1.

The processing pipeline was constructed of image segmentation, Hue-Saturation-Value (HSV) colour filtering and morphological operations designed to preprocess the image

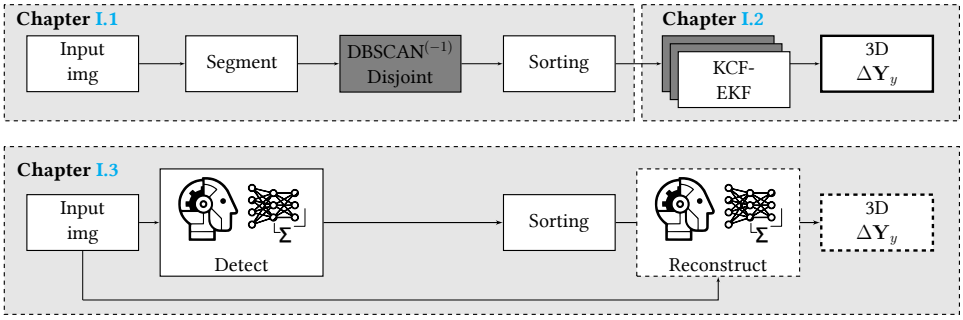


Figure IV.2.1: High level overview of the vision-based sensing approach proposed in Chapters I.1, I.2 and I.3.

frame for clustering operation. In the last step, the markers were detected and labelled with an unsupervised density-based clustering algorithm, DBSCAN [1]. The performance of DBSCAN was compared to another unsupervised clustering method, the Disjoint-set data structure [2] by processing (i) performance data obtained from a performance test with randomly generated cluster data and (ii) experimental data obtained from the measurements of the wing response undergoing oscillatory motion under gust excitations in the Open Jet Facility (OJF) wind tunnel at the Delft University of Technology.

The density-based approach, with DBSCAN, demonstrated more resilience against noise in the input image and higher accuracy. An error of ≈ 1 pixel was observed for most of the markers clustered with DBSCAN concerning the validation dataset. However, this Chapter highlighted an essential shortcoming of the de-noising HSV-morphological segmentation filter to clustering outcome. Namely, that (i) the clustering performance degraded without morphological operations and (ii) the mask for the clustering operations could be erased by morphological operations under certain lighting conditions (high illumination). This suggested that morphological operations are not always possible or beneficial, and highlighted the relevance and the need for a novel clustering approach.

To tackle this problem, a novel formulation of DBSCAN, the inverse DBSCAN^{-1}, was proposed, where the clustering problem is reformulated into a noise filtering problem. Instead of *rejecting* the noise, this approach *explicitly detects* the noise, making the clustering an *implicit* task. The experimental dataset was processed using the DBSCAN^{-1} pipeline, and it was shown that the actual clusters were successfully identified and isolated from the noise in the image. After isolating the clusters, DBSCAN^{-1} must be followed by an additional nominal DBSCAN clustering to extract the exact location of the markers. The final nominal DBSCAN can be done at a significantly lower computational cost due to the removed noise. Further studies were suggested to assess the performance gain of DBSCAN^{-1} compared to additional filtering steps in various lighting conditions.

Finally, the experimental dataset, consisting of image sequences, was processed with the clustering pipeline and analysed in the time and frequency domain. The main objective of this analysis was to understand whether the proposed image tracking method could correctly (i) estimate the time response and (ii) identify the frequency content of the measured wing displacement signal compared to the reference measurement provided by the laser vibrometer. For frequency domain analysis, the power spectral density of

the output signal was used as an assessment metric. The results demonstrated that the proposed method could capture and analyse the motion of an oscillating wing with relatively low-resolution cameras (1.3 megapixels). For this purpose, markers exhibiting high motion amplitude (closer to the tip) were the most favourable.

Main findings of Part I, Chapter I.1:

- Robustness against the noise of density-based clustering can be increased by reformulating the clustering problem into a noise filtering problem. The clustering problem becomes an implicit task, where the noise is no longer rejected but explicitly detected.
- The motion of an oscillating wing can be captured and analysed with relatively low-resolution cameras (1.3 megapixels) using unsupervised clustering integrated into an image processing pipeline.

Chapter I.2: State Estimation and Real-Time Tracking

This Chapter followed up on the second part of the proposed visual sensing approach as shown in the diagram in Fig. IV.2.1. The primary purpose was to develop a robust state reconstruction approach, which would take as input a point correspondence pair detected from two image frames and reconstruct the tip deflection response in 3D induced by gust input on the wing. Sorting and detection were provided by the methodology introduced in Chapter I.1.

The proposed method consisted of a tracker pair composed of a purely visual filter – a high-speed Kernelised Correlation Filter (KCF) [3], paired with an Extended Kalman Filter (EKF)–, allowing robust estimation of a system exhibiting oscillatory motion under the presence of marker failure and occlusions. The KCF-EKF method was validated experimentally on a real-time image stream of a very flexible wing subjected to gust excitation in the OJF wind tunnel. The method is further improved with the expansion of the EKF to the Augmented Extended Kalman Filter (AEKF) form, where the uncertain system parameters are included in the Kalman Filter model, thereby estimating both the states and parameters online adaptively. The last stage of the reconstruction step is the triangulation with the Direct Linear Transformation (DLT) method and coordinate transformation needed to obtain the relative displacement from the static displacement under a given free stream velocity.

The laser vibrometer measured response (in meters) was reconstructed with the KCF-EKF (dark grey) pipeline shown in the top right of Fig. IV.2.1. The camera setup was calibrated to obtain the reconstructed displacements, and the corresponding KCF-EKF point pairs from two image streams were triangulated with the DLT method. The details of the triangulation approach are given in previous studies conducted within the scope of this dissertation with the DLT and the parallel tracking approach [4, 5]. The obtained 3D coordinates were then transformed to align the laser vibrometer measurement. Similar to the prior measurement, the baseline static displacement was subtracted to obtain the transient displacement of dy as indicated by the last block in the diagram in Fig. IV.2.1.

Analysis of the experimental data and the reference data of the wing tip displacement measured by the laser vibrometer system highlight the sensitivity of the wing response

to gust frequency, the gust vane angle and the free stream velocity, allowing a better understanding of the requirement for the visual tracking. The higher free stream velocity and larger gust vane angles induce higher oscillatory wing tip amplitudes. The lower frequency of the gust alters the response characteristics from the impulse to 1-cosine. The shift is found near 3 Hz, where, for narrower 1-cosine gusts, wing response tends to attain the characteristics of an impulse, and the energy is absorbed in a narrower time span by the wing, while at lower frequencies, the wing response follows the shape of the 1-cos gust. The sensitivity analysis shows that higher amplitude and faster changes in response, higher gust frequencies, higher free stream velocity and gust vane angles are more critical for visual tracking and control feedback.

The reconstructed motion of the wingtip markers from KCF-EKF tracking showed good agreement with the reference measurement provided by the vibrometer. Differences in the peaks of the motion amplitude were found of up to 3.5 mm in the most adverse conditions, which were suggested to be due to (i) the splining of the laser reference measurement and (ii) coordinate system transformation between the laser and camera setup and (iii) stereo calibration errors. Equivalent frequency and time-domain analysis, as introduced in Chapter I.1, conducted on the reconstructed response, showed that the resonance peak could be captured well with relatively low-resolution cameras. In particular, it was suggested that the tracking performance could be directly improved by (i) more rigorous calibration of the tracking volume, (ii) more accurate measurement of the orientation and location of the camera setup concerning the wing and the reference system, and (iii) tuning of the smoothening introduced by the Kalman Filter.

Main findings of Part I, Chapter I.2:

- Motion of a flexible wing subjected to gust excitation can be estimated robustly through visual tracking methods by adding information about its dynamics in the Kalman Filter model. The system states and parameters can be estimated simultaneously by augmenting the Kalman Filter model with unknown system parameters, increasing the model's adaptability.
- Visual sensing is a robust and non-invasive approach for reconstructing wing motion for control feedback, which can be realised with relatively low-cost camera hardware and a combination of real-time algorithms.

Chapter I.3: Vision-Based Deep Learning Methods

In the previous Chapter, the reconstruction of the wing shape was realised through successive and relatively elaborate processing steps. Chapter I.3 investigated how the introduction of artificial intelligence, and in particular deep learning from raw image data, could help simplify the processes involved in the proposed smart visual sensing approach or even potentially replace it entirely.

To explore the deep learning-based concept of *Smarter Visual Sensing*, a Deep Convolutional Neural Network (DCNN) is trained to perform a part of the sorting operation as shown in Fig. IV.2.1 and detect the orientation angle of extracted contour shape of the wing. Simultaneously, the Geometrical Reflectional Symmetry Detector (GeConv) algorithm proposes an efficient algorithm-based solution to the problem to compare the two

methods. Both approaches show excellent results in detecting symmetry and orientation. The Neural Network can achieve an angle error of up to 0.05 degrees without needing a contour shape as input. However, it does not easily generalise shapes deviating from the training set. The traditional computer vision approach excels at speed and is model-free, meaning a good generalisation can apply the algorithm to many types of shapes. However, it requires a processing step to get the contour points, while the DCNN operates on the raw pixels. The main disadvantage of the GeConv is that, while the orientation can be easily corrected, the head or tail of the object along a given symmetry axis is arbitrary, which does not occur with the deep learning-based method.

The crux of the concept was based on the idea that a DCNN could be trained to accept sequences of raw image data and perform either a regression or classification task. Given a sequence of image data correlated through time, its successive evaluation with the trained DCNN should predict the variable of interest, either the orientation of an image, the location of a marker, or the wing displacement. This approach could yield a significant simplification as not only a single camera stream would be required, but the image processing steps are also reduced significantly.

Main findings of Part I, Chapter I.3:

- Deep Convolutional Neural Networks (DCNN) can be trained to replace parts or all processing steps involved in smart visual sensing by performing regression or classification from raw image data.
- Combining time-correlated image data with DCNN offers the potential for wing state reconstruction using only a single image stream.

IV.2.1.2. Part II: Smart Design & Optimisation

In this Part, the central element of the active morphing is investigated, namely, the system itself. First, a novel distributed morphing design is proposed and analysed. Some of its shortcomings are highlighted, which are addressed in the subsequent Part. A numerical study follows this up by investigating the benefits of the distributed morphing design in the context of online shape optimisation. The proposed approach optimises the steady-state lift-to-drag ratio for a given target lift coefficient online through evolutionary-based optimisation coupled with a Radial Basis Function Neural Network (RBF-NN). The analysis is done in preparation for the wind tunnel test discussed in Part III.

Chapter II.1: Distributed Morphing Wing Design

This Chapter presented a novel distributed morphing concept integrated into a seamless, actively morphing wing demonstrator called SmartX-Alpha. The proposed concept allowed variation of lift distribution locally along the span and addressed the drawbacks of the initial Translation Induced Camber (TRIC) concept. A Fluid-Structure Interaction (FSI) optimisation framework was developed to produce the optimised laminated design of the morphing skin, taking into account the ply orientation, laminate thickness, laminate properties and actuation loads.

Fixed ± 45 degree ply orientation and gradual ply thickness dropping achieved the desired morphing flexibility and manufacturability. The introduction of local lift variation

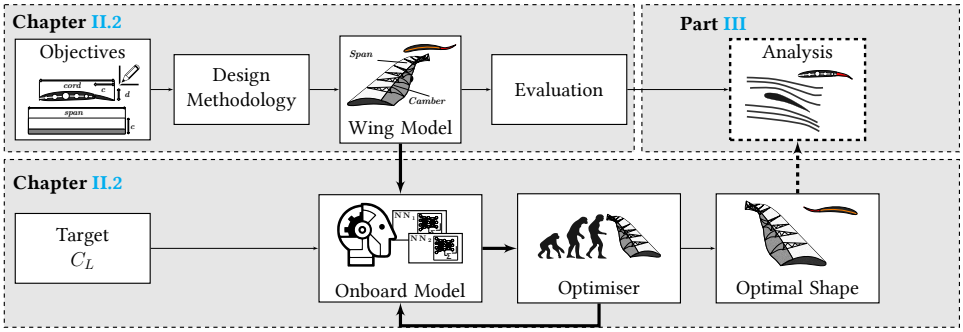


Figure IV.2.2: High level overview of morphing design and optimisation proposed in Chapters II.1, II.2.

presented an additional design challenge for inter-modular elastomeric skin design, which was required to allow sufficient flexibility and have the most negligible impact on the actuator load. A gap fully covered by silicone skin yielded the best design configuration for the inter-modular skin. The finalised concept is validated for the commanded target shapes with Digital Image Correlation (DIC) measurements.

The critical contributions of this Chapter are the realisation of the smart distributed actively morphing concept, characterisation and demonstration of the manufacturability of this concept. Another important finding of the validation study was the sensitivity to the backlash hysteresis. This nonlinear effect was identified as a mechanical imperfection resulting from manufacturing uncertainties and added stiffness due to the elastomer, causing lower than anticipated peak deflections. This finding, in turn, presented a worthwhile opportunity to investigate a possible control strategy designed to mitigate the impact of backlash through the learnings of previous sections involving novel hardware and sensor approach.

Main findings of Part II, Chapter II.1:

- Realisation of the smart distributed actively morphing concept, its characterisation and demonstration of its manufacturability.
- Revelation of the backlash phenomenon presents a key challenge to be solved utilising the capabilities of the smart actively morphing wing.

Chapter II.2: Learning-Based Aerodynamic Performance Optimisation – Numerical

Building further upon Chapter II.1, this Chapter investigated the aerodynamic performance improvement that could be achieved with the proposed seamless, distributed morphing design, utilising artificial intelligence as a black-box controller. Using lift and drag measurement updates, a novel learning-based shape optimisation method was proposed to optimise the shape online in order to maximise the steady-state lift-to-drag ratio for a given target lift coefficient. An online-trained Artificial Neural Network (ANN) acted as an onboard lift and drag coefficient value model for given actuator input angles, which could

be adapted online. An evolutionary optimisation algorithm was responsible for finding the most optimal shape and the angle of attack for the given lift coefficient targets. The sampling of pseudo-random inputs explored the optimisation space during the wandering phase. A surrogate model based on Radial Basis Function Neural Network (RBF-NN) was responsible for learning the mapping between wing shape and the angle of attack as inputs, and the lift and drag coefficient as outputs.

The simulation experiment demonstrated that the surrogate model could estimate sufficiently across a large spectrum of inputs and outputs, facilitating the optimiser to produce optimal wing shapes outperforming the jig shape in terms of aerodynamic efficiency. With the global nature of the optimisation approach, the optimisation strategy allowed to find wing shape and angle-of-attack combinations with lift-to-drag ratio increases of up to 14.6 % for a wide range of target lift coefficients without requiring further exploration. The ability to update the onboard RBF-NN surrogate model using new measurements demonstrated a relevant and beneficial aspect of the proposed approach to actively morphing systems, namely, *adaptability*. As demonstrated through the wandering phase, this approach would lend itself practical for the in-flight flight-test scenario, where the behaviour of the wing model is partly unknown and must be identified during manoeuvres. Furthermore, as the model is input-output based, the structure could be adapted to include other variables such as Reynolds and Mach numbers to cover the entire flight envelope.

Main findings of Part II, Chapter II.2:

- Evaluation of a black-box performance optimisation strategy can be adapted online and utilised for in-flight shape optimisation for an actively morphing wing.
- Global optimisation can be pursued by adaptive Neural Network structure based on Radial Basis Functions and evolutionary-based optimiser given sufficient knowledge of the optimisation space through exploration. Adaptability is a key aspect of an actively morphing system.

IV.2.1.3. Part III: Smart Control & Integration

In the previous Part, a novel morphing wing design was proposed, and a numerical study was conducted to investigate online aerodynamic performance optimisation. Following up on the previous one, this Part covers the fundamental control design required for achieving the main objectives and the integration of all systems – evaluated during wind tunnel experiments. First, a robust control method is proposed for simultaneous gust and manoeuvre load alleviation based on Nonlinear Dynamic Inversion (INDI) and Quadratic Programming (QP), tested experimentally in the OJF wind tunnel equipped with a gust generator. This is followed by a data-driven Incremental Model-based Nonlinear Dynamic Inversion (IM-NDI) control approach designed to mitigate the mechanical imperfections in a seamless active morphing wing, as established in Part II. Finally, this Chapter is followed by the development and validation of the complex experimental setup, where all previous elements of an active morphing wing are combined to investigate online performance optimisation in a real-life experiment.

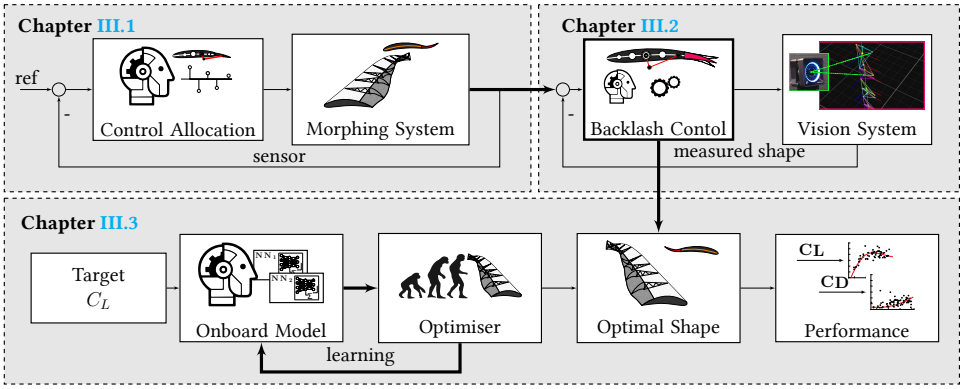


Figure IV.2.3: High level overview of control and online shape optimisation proposed in Chapters III.1, III.2 and III.3.

Chapter III.1: Gust and Manoeuvre Load Alleviation of a Morphing Wing

This Chapter investigates the design and wind tunnel testing of a simultaneous gust (GLA) and Manoeuvre Load Alleviation (MLA) control law for a seamless active morphing wing. First, the Incremental Non-linear Dynamic Inversion (INDI) is introduced for a generic nonlinear system. Since the morphing system has a distributed array of actuators, a control allocation strategy is investigated for the controller, which considers the actuator position, rate and relative position constraints. The proposed approach with INDI combined with Quadratic Programming (INDI-QP) satisfies these constraints in contrast to INDI with Pseudo Inverse control allocation (INDI-PI). To increase the smoothness of the control allocation, the INDI-QP is augmented with the virtual shape functions (abbreviated to INDI-QP-V).

The effectiveness of the proposed INDI-QP-V is validated by wind tunnel experiments in a complex experimental setup running in real-time. In manoeuvre load alleviation tasks, INDI-QP-V increased the total lift for performing pull-up manoeuvres without amplifying the wing root bending moment. In the presence of successive "1-cosine" gusts, it mitigated the loads without requiring any gust information. A key revelation of the proposed method was the robustness against aerodynamic uncertainties, gusts, actuator faults, and nonlinear backlash. The INDI-QP-V was effective in mitigating the effect of gust, while the 9^{th} actuator had become inoperable due to mechanical failure.

The evaluation of the proposed method also highlighted a critical factor responsible for the degradation of the load alleviation performance at higher gust frequencies, namely the phase lag in the closed-loop system. The lag is contributed by two sources in the morphing system, the identified cut-off frequency of the servo (2.60 Hz) and the filter's cut-off frequency needed to attenuate the relatively high noise level of the load balance system. Therefore, these aspects of the morphing system are considered limiting performance factors and must be designed accordingly to meet the desired performance criteria of the load alleviation objective.

Main findings of Part III, Chapter III.1:

- The proposed INDI-QP-V control is an efficient and easy-to-implement sensor-based control strategy for simultaneous gust and manoeuvre load alleviation of an over-actuated and over-sensed actively morphing wing.
- The ability to account for model uncertainty through real-time control and sensing architecture has demonstrated that objectives can still be met in a fault-tolerant manner, even in the presence of significant model error sources such as actuator fault and backlash. This is an important step toward the scalability of the concept to full-scale test flight.
- The responsiveness of the servo and sensor strategy is the limiting factor for faster objectives of an actively morphing wing.

Chapter III.2: Mitigating Backlash and Disturbances Using Vision-Based Control

This Chapter presents the experimental verification of a data-driven Incremental Model-based Nonlinear Dynamic Inversion (IM-NDI) control approach designed to mitigate the mechanical imperfections in a seamless active morphing wing. The system dynamics are identified online using the stored input/output data without a prior-known model. Then the NDI controller is developed based on the identified dynamics.

A crucial aspect of improving these imperfections is an accurate knowledge of the morphing wings' shape. A vision-based control system was developed, which has shown to be adequately effective for this task, given its robustness, high frame rates (250 frames per second), and good calibration accuracy (reprojection error 0.25mm).

A real-world experiment was conducted based on computer vision feedback to evaluate the proposed method. The experimental results demonstrate that the morphing wing can track reference signals with different frequencies despite external disturbances by applying the IM-NDI. Under Feed-Forward (FF) control, the morphing wing suffers from mechanical imperfections, reflected by the tracking responses' lagging and magnitude shrinking phenomena. The performance of Feed-Forward control also degrades in the presence of external disturbances. By contrast, experimental results show that IM-NDI can effectively decrease the tracking errors by more than 62 % despite disturbances.

Furthermore, the proposed vision-based system combined with the control methodology demonstrates the ability to compensate for mechanical imperfections without changing the morphing hardware. All results collectively illustrate the effectiveness of the proposed IM-NDI method in dealing with inherent mechanical imperfections of the morphing wing system.

Main findings of Part III, Chapter III.2:

- The adaptability of the proposed vision-based IM-INDI control approach leads to better robustness against disturbance and backlash and is able to cope with mechanical imperfections in the morphing system.

- Limited servo bandwidth impairs the ability to mitigate disturbances and track faster-referred signals with compensating controllers.
- The effectiveness of backlash compensation can be achieved without the requirement for invasive changes to the design of the morphing mechanism.

Chapter III.3: Learning-Based Aerodynamic Performance Optimisation – Experimental

Expanding on the objectives achieved experimentally in the previous Chapter, namely, simultaneous gust and load alleviation, this Chapter focuses on the realisation of the shape optimisation objective for improved performance of the morphing wing. Moreover, this Chapter builds further on the simulation study presented in Chapter II.2 and proposes and validates an online black-box shape optimisation architecture experimentally for distributed actively morphing wings.

The proposed approach consists of an evolutionary optimisation strategy, Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) coupled with a low cost, Radial Basis Function Neural Network (RBF-NN) onboard surrogate model for online optimisation of the steady-state lift-to-drag ratio of the morphing wing for a given target lift coefficient. The onboard RBF-NN model is trained with experimental data collected during wandering phases to capture the behaviour of the wing model and predict the steady-state lift and drag coefficients. The computational time is reduced by implementing a parallel query of 150 candidate solutions for the angle of attack and virtual shape using CMA-ES in the inner optimisation loop. The RBF-NN model is updated globally using the outer loop, based on actual measurements.

Compared to the unmorphed wing base shape, a drag reduction of 7.8 % was achieved on the SmartX-Alpha demonstrator for a target lift coefficient of 0.65. For a wide range of target lift coefficients, the predicted drag reductions vary between 6.5 % and 19.8 %, with higher drag reductions associated with lower lift coefficients. Although shape optimisation is governed by the steady-state behaviour of the wing, accurate and fast shape feedback is a crucial factor for mitigating the backlash effects and providing accurate shape feedback such that the online model can be adapted. Vision-based control and sensing are critical factors for achieving this in an effective and non-invasive manner.

The proposed approach has shown significant benefits over existing methods. Firstly, grey-box methods rely on model excitation manoeuvres at every trim condition, which is undesired. The proposed methods can eliminate this need by employing a black-box approach and online adaptation, providing the optimal wing shape for various lift coefficients. The black-box nature of the onboard model facilitates the use of experimental or in-flight data to capture the wing model behaviour. Secondly, the black-box nature of the proposed methods is model-independent, meaning that the strategy can be applied to other morphing wing platforms and concepts with adequate learning. This, again, is a critical step towards maturing the active morphing wing technology and conducting a full-scale test flight. Finally, the current optimisation structure allows the implementation and development of other optimisers which can be implemented with the same black-box data structure in the inner loop.

Main finding of Part III, Chapter III.3:

- The proposed black-box online shape optimisation approach demonstrates significant benefit over grey-box methods, as no excitation is required at every trim condition. The black-box nature of the onboard model facilitates the use of in-flight or experimental data to capture the wing model behaviour. Moreover, this approach is model-independent and scalable to generic morphing wing platforms due to its adaptability.
- High-accuracy and fast sampling sensors are critical for backlash compensation and shape optimisation in an experimental setting. Vision-based sensing is a critical, non-invasive sensory element for active morphing systems that rely on shape feedback.

IV.2.1.4. Part IV: Outlook

The active morphing wing demonstrator concept was proposed in the previous sections, and its feasibility was investigated through incremental improvements and successive wind tunnel tests.

A key finding was that actuator bandwidth, and the quality of the sensor noise attenuation were critical factors required to push the envelope for multi-objective optimisation and mitigation of aeroelastic loads. The last Chapter investigates the impact of faster actuation on load alleviation performance and the ability to achieve faster aeroelastic objectives. For this purpose, a distributed over-actuated aeroelastic wing demonstrator was developed, the SmartX-Neo, allowing faster actuation with a much simpler flap mechanism and ensuring the same integrated sensing capability as the SmartX-Alpha. The difference in objectives addressed with the two demonstrators is shown in Fig. IV.2.4 below.

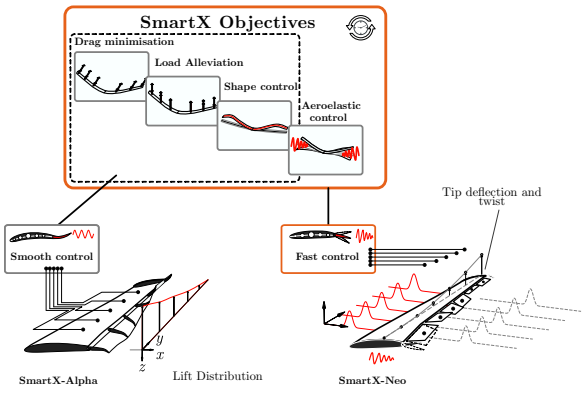


Figure IV.2.4: Comparison of the objectives of the SmartX-Alpha and SmartX-Neo.

The effect of actuator design was studied through a series of gust simulations of closed-loop control of a parametric aeroservoelastic model of the SmartX-Neo, for gusts with various frequencies. Actuator dynamics were implemented in the system by scaling the

stiffness and damping of a second-order system. With the nominal Linear Quadratic Regulator (LQR) controller, reductions of peak gust load up to 78 % were achieved compared to the open-loop case. It was observed that lower scaling corresponding to faster actuation provides significant improvements of up to 11 per cent over the nominal actuator configuration, yielding reductions of gust loads up to 84 %. Furthermore, it was observed that the effectiveness of faster actuators improves for higher frequency gusts, confirming the potential of SmartX-Neo to deal with faster control objectives more effectively.

This Chapter's main contribution is a step towards maturing and increasing the Technology Readiness Level (TRL) of distributed morphing and over-actuated wing concepts, which is a critical step towards adopting this technology in future aircraft designs.

Main finding of Part IV, Chapter IV.1:

- For an over-actuated aeroelastic system, a faster actuator can be represented by a proportional scaling value below one and yields better load alleviation and improved effectiveness for higher frequency gusts.
- Realisation of a distributed-over actuated aeroservoelastic wing concept through a simpler and faster actuation mechanism.

IV.2.2. Recommendations

This section presents the recommendations of this dissertation and a discussion on the scalability of the proposed technologies.

IV.2.2.1. Maturing Wing Morphing Technology

The final Chapter, Chapter IV.1, sets an important step towards maturing active morphing wing technology with the realisation of a simple over-actuated wing concept, extending the capabilities of a distributed smooth morphing wing with a simplified actuation approach. The implicit aim of this dissertation, to accelerate the TRL and the adoption of actively morphing wing concepts, which must include a smart sensing, control and integrated system, is therefore strengthened through the findings in this dissertation. Still, some recommendations can be made on the specifics of an actively morphing system presented in previous Chapters and assumptions made in the final Chapter.

The numerical experiment, performed in Chapter IV.1, presents a purely numerical scenario with the assumption of full state feedback of the system. In a more realistic case, a state estimator must be integscrated with the LQR (analogous to Linear Quadratic Gaussian (LQG)). Moreover, the noise and bandwidth limitations of the sensors will impact the effectiveness of Gust Load Alleviation (GLA). Including and studying these effects in further assessment of fast actuators within the scope of GLA and MLA is recommended. Furthermore, it is recommended to evaluate the prioritisation of objectives on the effectiveness of gust load alleviation to include the impact of specific control allocation on drag performance.

IV.2.2.2. Smart Visual-Based Sensing

In this dissertation, a non-invasive vision-based sensing strategy was proposed to estimate the shape of a morphing and flexible wing with distributed control surfaces in real-time, suitable for control feedback. Model independence and non-invasiveness were demonstrated by incremental improvements to the vision-based sensor system across various closed-loop control experiments. Several critical elements of the processing pipeline were investigated: unsupervised clustering, correlation-based tracking filtering, Kalman Filtering, 3D reconstruction and deep learning.

Adaptive Clustering for Increased Robustness

The proposed formulation of DBSCAN, the inverse DBSCAN (DBSCAN^{-1}), demonstrated how the clustering problem is reformulated into a noise filtering problem, where, instead of rejecting it, the noise was detected explicitly, making the unsupervised clustering an implicit task. The image segmentation analysis also showed that the approach chosen for image thresholding plays a critical role in obtaining an optimal mask for the clustering problem. Various approaches were investigated, using global normalisation, baseline thresholding and Otsu's thresholding, where Otsu's method produced the cleanest mask. However, the latter method still assumes that the image pixel intensity histogram has bimodal nature (i.e. there is a clear separation between an image background and foreground), which may not always be the case, in particular, with a noisy dataset.

It is evident that there is a significant link between segmentation and clustering. However, since pixel intensities naturally reflect a dynamic response captured from the oscillating wing, these processes must also be connected in time. Implementing a Sliding Discrete Fourier Transform (SDFT) could yield an efficient method for capturing the change in the pixel intensities or its *footprint* in the spatial domain [6]. The thresholding value for the segmentation can be adjusted adaptively across the image sequence. As more data reflecting the motion of the marker and its corresponding intensity is collected gradually in the SDFT time window, a distinct value of the resonant frequency of wing motion can be obtained such that the threshold value is adjusted. The suggested approach is illustrated in Fig. I.1.31.

The accuracy and motion resolution could be further improved by increasing the camera's resolution and utilising markers less sensitive to lighting conditions, such as active Infrared (IR) markers. However, a thorough trade-off is essential, as increased resolution would generally increase the computational cost of the tracking pipeline and reduce the maximum processing frame rate, subsequently reducing the tracking bandwidth.

Tracking Dynamics and Reconstruction

This dissertation proposed a reconstruction method combined with a 2D tracking method followed by a 3D reconstruction step from synchronised image frames. The proposed KCF-EKF tracking method demonstrated that significant tracking speed and robustness could be achieved when a purely visual filter is paired with an Extended Kalman Filter, allowing system prediction through knowledge of its dynamics. Since the Kalman Filter was coupled to the motion of a marker in a 2D image plane, the simplest representation of the system is linear particle dynamics. The dynamics can be further enhanced by augmenting the Kalman Filter model with system parameters, allowing state and parameter estimation. As the wing exhibits oscillatory motion under gust excitation, which is also true in the 2D

image plane, oscillator dynamics were used in the augmented formulation of the AEKF. This formulation did not take into account disturbance to the system, which was caused by a discrete gust input. The gust input can be modelled as an unknown disturbance force and estimated together with the remainder of the parameters. The structure of the AEKF can expand the estimator dynamics to more complex dynamics, further improving the estimation.

It must be noted that in the proposed approach, the Kalman Filters estimating a corresponding image pair from two camera streams are not coupled. Using this approach, a strategy can be devised where the two Kalman Filters are coupled through a synchronisation variable, indicating the change of state in either position or the marker's velocity. Finally, the robustness of the reconstructed 3D estimate of the correspondence pair can be further improved by including a Kalman Filter estimate of the 3D position and velocity. This approach has already been demonstrated in a study conducted in a mobile experimental setup [5]. Finally, the computational cost of the preprocessing and tracking can be further reduced by utilising a parallel tracking approach with Threading Building Blocks (TBB) as developed in the previous study conducted within the scope of this dissertation.

Deep Learning

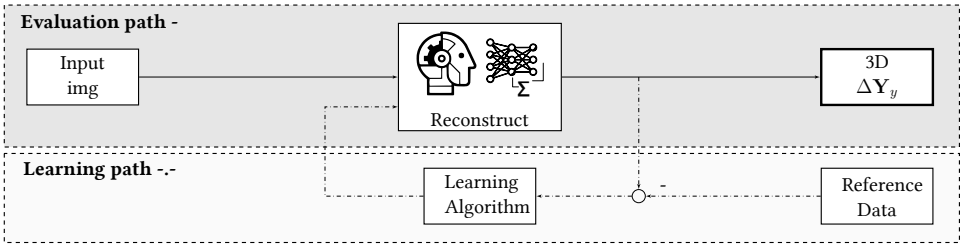


Figure IV.2.5: 3D Reconstruction pipeline replaced fully by Deep Learning approach.

Chapter I.3 demonstrated that the relatively elaborate image processing steps in Chapters I.1 and I.2 could be replaced in part by the deep learning approach from raw image data. A DCNN was trained to perform operations relating to the sorting and clustering of images. To further simplify the process, the entire reconstruction can be potentially replaced by DCNN. In this potential setup, as shown in Fig. IV.2.6, a DCNN can be trained to predict the 3D marker displacement from features learned in the raw image data. Feeding the sequences of images would then yield a lookback prediction of the current displacement. This would be a significant simplification as not only a single image is used, meaning a single camera image would suffice, but all the processing steps would also be removed.

Developments in the methodologies and hardware have significantly improved the capabilities of artificial intelligence (AI) and data-driven sensing, allowing for real-time implementation of control tasks. With the recent developments in Recurrent Neural Networks (RNN) [7] and Graph Neural Networks (GNN) [8], the ability to introduce time dependency in the Convolutional Neural Network (CNN), which is tailored for image-based spatial data, becomes a promising aspect. The proposed approach, where image sequences are not correlated, can be improved in two ways. First, a recurrent network structure can be considered, where the image sequences are correlated and thus better reflect the time

history of the wing deflection. Secondly, while training the neural network, the input channel can be blended with other time-series data to represent a spatio-temporal image fusion, as shown in Fig. IV.2.5. It must be noted that there are many challenges and caveats to such an approach as unbalanced data, noise and domain adaptation can be significant issues for successful deployment of such DL-based methods. Therefore, further investigation is necessary to assess the potential of these types of methods.

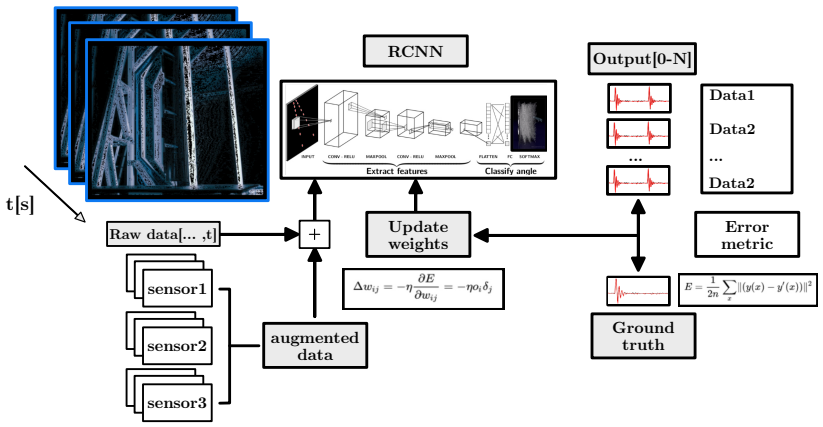


Figure IV.2.6: General overview of the DCNN structure and feature extraction process.

IV.2.2.3. Morphing Wing Design & Optimisation

The distributed active morphing design concept proposed in this dissertation aimed to address the drawbacks of previous morphing designs, and its manufacturability was demonstrated through the realisation of the hardware demonstrator.

Design Improvements and Challenges

The morphing modules' Symmetric Bend Up and Down actuation showed a linear trend, agreeing with the simulation. However, the validation study revealed a significant sensitivity to the backlash phenomenon and added stiffness due to the addition of the elastomer, leading to the overestimation of peak deflections. While model correction from reference data yielded a better estimation, the numerical model can be further improved by implementing greater detail of the actuation components and interactions of the interfaces (e.g.the guiding lip and servo pickup). However, a trade-off must be made between model complexity (i.e. high computational cost) required to capture the combined effect of backlash in the actuation mechanism and added stiffness of the elastomer and the achievable accuracy.

Capturing and integrating these effects in the design process cost-efficiently presents a challenge worth investigating in further studies. Mitigating the backlash effect opens another opportunity for research exploring the current concept. In Chapter III.2, a closed-loop control system, relying on accurate position measurements from the vision-based

sensors, demonstrated how this effect could be mitigated effectively by a data-driven IM-NDI control approach. Finally, a recommended future research topic is to investigate the impact of a more flexible wing structure in torsion and span-wise bending.

Learning-Based Optimisation Structure

Combing the potential of the proposed distributed morphing concept with artificial intelligence, an investigation of feasible aerodynamic performance improvement was made by implementing a novel black-box controller and evolutionary optimisation strategy. With this controller, the shape of the wing was actively optimised for a given target lift coefficient using the lift and drag measurement updates. The simulation experiment demonstrated that the optimisation strategy allowed optimal wing shapes to outperform the jig shape in terms of aerodynamic efficiency for a wide range of lift coefficient targets while also allowing the onboard RBF-NN surrogate model to be adapted online. The input of the onboard model consisted only of the wing shape and angle of attack, and the outputs, the lift and drag coefficients. Aside from these inputs, the Reynolds number and Mach number have a prominent influence on lift and drag coefficients. Consequently, the input space of the onboard model should be expanded to include these parameters across the entire flight envelope. With adequate training, the RBF-NN can be easily suited to this purpose owing to its black-box nature.

Due to the adaptable onboard model and its flexible inner-outer-loop structure, the online optimisation framework presented in Chapters II.2 and III.3, allows the implementation and development of other optimisers, which can be integrated using the same neural network model in the inner loop. However, it is recommended to investigate Hyperparameter Optimisation (HPO) strategies for the neural network and optimisation parameters to enhance further the prediction performance of the RBF-NN model.

IV.2.2.4. Control & Integration of Morphing Wing Systems

Following the design and numerical studies with the proposed concept, a robust control method was proposed for simultaneous gust and manoeuvre load alleviation and tested in a wind tunnel experiment. This was followed by the development and testing of a complex experimental system. Here, all previous elements of an active morphing wing were combined to investigate black-box online performance optimisation in a real-life experiment.

The evaluation of the INDI-QP proposed method highlighted the phase lag in the closed-loop system as a critical factor responsible for the degradation of the load alleviation performance at higher gust frequencies. Contributed to the identified cut-off frequency of the servo and cut-off frequency of the filter needed to attenuate the relatively high noise level of the load balance system, the apparent approach to improving the performance is to make the servo faster and the sensor less noisy.

The final Chapter of this dissertation, Chapter IV.1, demonstrated that faster actuation significantly improves the ability to cope with faster objectives and the effectiveness of gust load alleviation. The modularity of the distributed TRIC concept and conventionally driven actuation system allows selecting the best available servo within the cost limits and performance requirements. Dividing the wing section into smaller modules is another approach to reduce actuation and increase the actuation speed. A different approach is to

replace the relative complex morphing modules with a simply hinged distributed array of discrete flaps. The attainable wing shape, hence lifts distribution, would be less smooth; however, the actuation loads are significantly reduced, and the skin is no longer bent or twisted for actuation. SmartX-Neo was initiated to investigate the benefits of this approach. Further tests are recommended to assess the potential of these two distributed over-actuated wing concepts.

Sensor setup can be improved by relying on less noisy sensory information or developing a sensor fusion strategy from various sensor sources. The information about the upstream gust can also be observed and disclosed to the controller with Light Detection and Ranging (LIDAR) based sensors. Several options have been proposed in the scope of sensor fusion, which includes centralised, parallel, decentralised, and cascaded Kalman Filters [9]. A decentralised Kalman Filter is a suitable approach when the system is distributed; a large number is sensor data is needed to be processed quickly, and fault detection is desired [10]. A potential sensor diagram is shown in Fig. IV.2.7 with a decentralised setup.

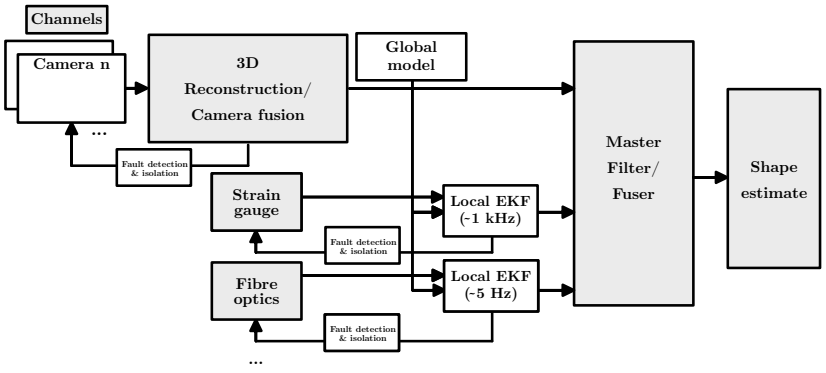


Figure IV.2.7: Conventional sensor fusion approach with Decentralised Kalman Filtering.

Selecting the optimal Kalman Filter configuration is the most challenging aspect of this setup, and thorough characterisation of the sensor's performance, calibration and assessment of uncertainties is required for the multitude of sensors. This master fuser then has the ultimate decision to merge the sensor data accordingly such that the amount of uncertainty in the final output is reduced. The output can be formulated to include the switching modes or adaptive modes to prioritise certain types of sensors during a specific control task (e.g., GLA, MLA). Furthermore, adaptability can be added to detect faults and adjust sensor mixing.

IV.2.2.5. In-Flight Performance Optimisation and Architecture

The final experiment in Chapter III.3 demonstrated the realisation of the black-box learning-based shape optimisation approach for maximising the steady-state lift-to-drag ratio for a given target lift coefficient using the lift and drag measurement updates. The approach was evaluated using a complex experimental control architecture. An additional layer of

vision-based control was responsible for backlash compensation using real-time shape updates from the vision system. The realisation of the complex experiment is an essential step toward in-flight shape optimisation for active morphing wings with various layers of control and sensing components and objectives.

The ability of the proposed architecture to realise the best wing shape for various lift coefficients online eliminates the need for model excitation manoeuvres at every trim condition, as is characteristic of existing grey-box methods employing local models. Moreover, as discussed previously, the onboard model can and should be expanded to include other input parameters characterising the flight envelope. The adaptability, model-independency and non-requirement of model excitation are key benefits of successful deployment of the black-box in-flight optimisation approach. Here again, the use of HPO can help accelerate the expansion of the model input space and adaptation to in-flight data.

IV.2.2.6. Scalability of Proposed Technologies

As with most morphing concepts and supporting technologies, scalability to full-scale aircraft is the hardest and, for particular concepts, an infeasible challenge. However, this is *the* critical step needed for maturing the technology for adoption in the aircraft industry. Some of the known reasons include (i) coupling of the actuation approach with not easily scalable smart materials (e.g. piezoelectric materials), (ii) power and control system demand (e.g. high voltage and amplifiers), (iii) low control authority, hysteresis and nonlinear behaviour (e.g. actuation, temperature sensitivity), (iv) weight impact due to actuation and control system and finally, (v) maintenance and fatigue of the proposed system.

Several observations were made in the previous sections regarding the choices and merits of the technology proposed in this dissertation. Therefore, the choice was made in this dissertation to consider the holistic view of the active morphing wing system. The latter is represented by the building block analogy, each with associated technology which is easy to integrate but is detachable from a particular choice for the morphing wing concept.

In Part II, distributed actively morphing concept was realised, and its manufacturability was demonstrated through the implementation of conventional actuation and manufacturing methods. The inherent simplicity of the distributed TRIC concept, low actuation forces and modular principle are favourable aspects for upscaling the morphing technology. While adapting the concept to full-scale aircraft will not come without additional obstacles, enabling larger wing surfaces and coping with higher control surface loads can be supported by larger conventional actuators and adequate design of the supporting composite structure. However, as highlighted in Chapter II.1, the inherent principle of distributed TRIC requires chordwise actuation and spanwise motion occurring at the sliding interface, which comes with associated mechanical friction contributing to mechanical imperfections. The latter is further exacerbated by the added stiffness of the elastomer, but can be mitigated to an extent by a rigorous selection of materials and designs.

However, an important aspect to consider is potential fatigue and debonding of the skin at the upper and lower skin bondline (e.g. at the trailing edge or leading edge bond or the composite layup near the highly loaded skin plies) or the interfaces with the actuation mechanisms and the elastomer connection. The distributed approach is beneficial due to its modularity and replaceability. However, routine maintenance is required to monitor

the wear of the mechanical components and additional interfaces. Consequently, consideration must be made to improve the bond interface near the leading edge, trailing edge and elastomer connection. Furthermore, the fatigue characteristics of the composite skin in high strain areas (the upper/lower surface near the wing box-flap) must be assessed. Collectively, these considerations are critical for convincing the industry to adopt active morphing concepts and lay the foundations for the certification process.

As presented in Part I, the proposed sensing technology relies only on external visual markers or raw pixel data, which is adaptable to the chosen system (wing). As demonstrated in Chapters I.1 and III.3, it was equally effective for the forward swept, flexible wing as the distributed morphing wing. Consequently, it was developed with scalability in mind.

As presented in Part III (Chapter III.1), the simultaneous gust and manoeuvre load alleviation technology, based on an incremental sensor-based control approach, demonstrated the ease of implementation, robustness to actuator fault and backlash in a real-life experiment with a relatively low computational load. These traits make the proposed approach a favourable candidate for alleviating excessive loads in full-scale flight caused by manoeuvres and gusts. Manufacturing and mechanical imperfections, the consequent model and sensor uncertainty and system complexity are expected to play a more prominent role in upscaling of the wing and transition to full-scale flight. The ability to cope with model uncertainties and external disturbances, presented in both Chapter III.1 as III.2, without the need for invasive changes to the design of the morphing mechanism, is similarly favourable merit for scalability.

Finally, the online black-box shape optimisation architecture proposed in Chapter III.3 shows promising potential for in-flight application to generic morphing wing platforms. This potential was emphasised by the ability of the online RBFNN onboard model to generalise to a secondary wing model with different aerodynamic baseline shapes. The inherent ability to restructure the onboard model for new input space and training data from data-driven sources (i.e. numerically or experimentally generated) is a significant merit for the scalability and generalisation of this technology.

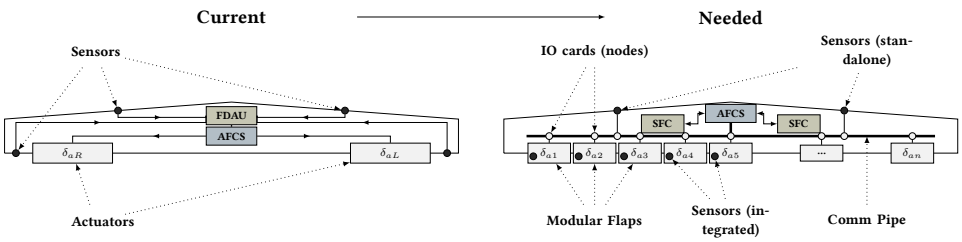


Figure IV.2.8: Differences conventional and distributed Real-time framework and synchronisation for control and online data processing.

Distributed control and vision-based sensing architecture were critical elements of shape optimisation, which were implemented non-invasively through incremental improvements of the hardware demonstrator across various experiments. The final recommendation for future applications to real-world commercial aircraft is to deploy this approach in scaled test-flight and integrate Graphics Processing Unit (GPU) based hardware-

acceleration for online processing and fusion of the sensory data in unison with Flight Test Instrumentation (FTI) and Sensor Fusion Computer (SFC). An illustration of the approach, analogous in configuration to the SmartX-Neo presented in Part IV, Chapter III.3 is shown in Fig. IV.2.8. The hardware acceleration can be achieved with embedded GPU computational units, such as the Jetson AGX TX2/Xavier and real-time systems target machines, such as the Speedgoat [11, 12].

References

[1] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, Vol. 96, 1996, pp. 226–231.

[2] Galler, B. A. and Fisher, M. J., "An improved equivalence algorithm," *Communications of the ACM*, Vol. 7, No. 5, may 1964, pp. 301–303.

[3] Henriques, J. F., Caseiro, R., Martins, P., and Batista, J., "High-Speed Tracking with Kernelized Correlation Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, No. 3, mar 2015, pp. 583–596.

[4] Mkhoyan, T., de Visser, C. C., and De Breuker, R., "Parallel Real-Time Tracking and 3D Reconstruction with TBB for Intelligent Control and Smart Sensing Framework," *AIAA Scitech 2020 Forum*, American Institute of Aeronautics and Astronautics (AIAA), jan 2020, p. 2252.

[5] Mkhoyan, T., "tmkhoyan/parallelTrackingTBB: initial release," dec 2019.

[6] Jacobsen, E. and Lyons, R., "The sliding DFT," *IEEE Signal Processing Magazine*, Vol. 20, No. 2, mar 2003, pp. 74–80.

[7] Liu, Y. and Hansen, M., "Predicting Aircraft Trajectories: A Deep Generative Convolutional Recurrent Neural Networks Approach," Tech. Rep. arXiv:1812.11670, arXiv, Dec. 2018.

[8] Wan, S., Gong, C., Zhong, P., Du, B., Zhang, L., and Yang, J., "Multiscale Dynamic Graph Convolutional Network for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 58, No. 5, May 2020, pp. 3162–3177.

[9] Sijs, J., Lazar, M., van den Bosch, P., and Papp, Z., "An overview of non-centralized Kalman filters," *2008 IEEE International Conference on Control Applications*, IEEE, San Antonio, TX, USA, 2008, pp. 739–744.

[10] Sun, S.-L. and Deng, Z.-L., "Multi-sensor optimal information fusion Kalman filter," *Automatica*, Vol. 40, No. 6, June 2004, pp. 1017–1023.

[11] NVIDIA, "High Performance AI at the Edge | NVIDIA Jetson TX2," .

[12] Speedgoat, "Aerospace Industry Solutions | Speedgoat," .

Appendices

IV.A1

Planform Parameters

Table IV.A1.1: Planform parameters.

Parameter		Symbol	Value	Unit
Wing aspect ratio		$\mathcal{R}$	13.6	[-]
Wing span		b_w	3.4	[m]
Wing loading		b_w	6.706	[kg/m^2]
Fuselage length		L_f	2.345	[m]
Elevator span		b_e	0.56	[m]
Vertical stabilizer span		b_s	0.25	[m]
Wing chord	(NACA 0015)	c_w	0.25	[m]
Elevator chord (root,tip)	(NACA 0010)	c_e	(1.67,1.45)	[m]
Vertical stabilizer chord (root,tip)	(NACA 0010)	c_s	(0.24,0.19)	[m]
Elevator incidence angle		α_i	-4	[$\circ$]
Fuselage position	x (w.r.t. wing)	x_f	-0.645	[m]
c.g. position	x (w.r.t. wing)	x_{cg}	0.027	[m]
Neutral point	x (w.r.t. wing)	x_{np}	0.305	[m]
Total mass		W_t	5.7	[kg]
Payload mass		W_p	1.5	[kg]

Acknowledgements

One can characterise a PhD journey in numerous ways. When *time* is the axis reflecting its progression, this journey should never be time-invariant of nature. In my experience, the dynamics involved here are better reflected by highly nonlinear dynamics subject to uncertainties and disturbances. However, inherent to this specific journey, every onset of uncertainty and disturbance is followed by growth through adaptation and perseverance. More importantly, uncertainty is balanced by the gift of reinforcement and stability provided by supporting colleagues, friends and family. Within this small section reserved in my otherwise very long (and hopefully not too tedious) dissertation, I would like to seize the opportunity to thank many people who played an important role in my journey.

First of all, I would like to thank my promotors, Roeland De Breuker and Coen de Visser.

Roeland, my first promotor, you were the one who presented me with the opportunity to pursue a PhD in a topic that had fascinated me for a long time. You did not only believe in me but always supported my ambitious ideas, no matter how eccentric or even crazy they might have been. As you later told me, giving me the freedom to explore the most enjoyable and captivating direction was important to the process of shaping me as a researcher. Thank you sincerely for that. Your trust and encouragement were the supporting pillars during my years as a PhD researcher, from start to finish. A journey that turned out to be the most challenging and rewarding thus far. I am thankful that I could always rely on your critical eye and razor-sharp mind, which have never ceased to amaze me.

Coen, your role as my second promotor was paramount to me and my work. Supporting me with your knowledge and pragmatism, you have guided me through the toughest and most rocky points of my journey. You are a person of vast knowledge on numerous topics, a fellow programmer who loves implementing complex ideas –a talent many do not possess. Your office was the space where any topic was up for debate, including my objections against the existence of Windows. This belief eventually commanded me to find a way to circumvent the admin restrictions and install my favourite flavour of Linux on my TU Delft PC. Wise as you were, you told me that there are use cases for applications "certain office applications", which I may realise further down my career path. Guess what my colleagues are using now at the office (*cry emoji).

Aside from my promotors, I would like to share my gratitude for having one colleague and a friend who was of paramount importance directly to my research and my development as a researcher. Sherry, you never stopped believing in me and supporting me when everything else failed; I am forever thankful to you. We fought many battles together in the wind tunnels and beyond. Drafting together the most complex ideas and realising them in the most complex experiments, with the most complex systems possible, while rebutting the most critical reviewers and solving numerous bugs constrained by impossible deadlines. You are among the most brilliant, if not most gifted, researchers I know. Without you, this journey would not be complete, and I am glad to have shared such a critical part of it with you.

The SmartX project would not have existed without its team and supervisors. I am very glad to have had the opportunity to collaborate with many people from different departments across ASM, AS&CM, NDT and NovAM, including my direct PhD team members, Nakash and Vincent. Particularly, Vincent – although we had very different working approaches and did not always agree, that contrast has always complemented our work as a team. My respect for you goes deep; having you as a colleague was a valuable experience.

Jurij, you were not only part of the supervisory team of SmartX but also my fellow office colleague, a vast source of knowledge for myself and many others in the project on an extensive range of topics. I truly enjoyed our discussions on and off-topic and the research we did together, sharing a common passion for fixing bugs and problems encountered during a typical experimental campaign.

My first acquaintance with one of the supervisors, Roger Groves, far precedes my PhD journey. Roger, I have had the pleasure of working with you first in my DSE and later as an assistant researcher at the NDT lab. Your support and supervision have helped me start my journey in academia as a researcher; I am forever thankful for that.

Pim Groen, you were there for a short time, but your positivity and pragmatism had a lasting impact on us all. You will be deeply missed.

Sybrand van der Zwaag, although you joined the supervisory team later, your role in the project has been crucial. In the project's most high-stakes, critical phase, we were confronted by looming travel restrictions at the start of the pandemic. You intervened, took immediate action, and arranged our formal approval to travel to Germany, manufacture the wing and return before the lockdown. Without your timely help, the wing demonstrator would never materialise, and we would most likely never have had the opportunity to do the wind tunnel tests. Aside from that, although we have not met on many occasions, I have rarely met someone with such a positive, knowledgeable and calming aura which unconsciously demands deep respect.

And, of course, special thanks to Dean Henri Werij for granting the travel approval and making the project possible in the first place.

Continuing, I want to acknowledge someone of exceptional skill and knowledge without whom we would most likely not have a wing today. Martin Weberschock, you made us feel at home in your workshop. Your exceptional craftsmanship helped steer the project towards success. First, with the SmartX-Alpha and later with the SmartX-Neo, it was my privilege to work with you and learn from you, not once but twice – an experience I will value forever.

Acknowledgement of those who have played supervising roles in my academic journey would not be complete without thanking one of my greatest role models, Max Mulder, who has continuously shaped me as a researcher throughout my time in Delft. Max, although you weren't technically part of my PhD, you will always be my mentor, and I will always remain your student. I know that once I step into your office, no matter how short the visit, I leave as a wiser man than when I entered. Thank you for your support and guidance and for sharing your brilliance and wisdom with me.

I am very privileged to have had the opportunity to supervise and work with other researchers and MSc students. In the order of graduation: Abdul, Nisarg and Oscar – I am extremely proud of you and happy to have been part of your MSc adventure. Abdul, you were one of the most hardworking students I have ever met. Your work on system identification and model reduction was very impressive.

Nisarg, it's hard to describe the value you have put into the realisation of the SmartX project and your input to my PhD journey. You were not only a critical part of the design of the morphing wing but also the critical hand during the birth of the SmartX-Alpha. I enjoyed the time we travelled together in Germany, got our hands full of epoxy and inhaled sweet composite nano dust. You impressed me with your design knowledge, knowledge of composites, manufacturing, analysis and life in general. I am glad I was also part of your journey, as you were part of my own.

Oscar, Bo, Sherry and Iren, my rockstar wind tunnel warriors. Oscar, you were part of the last part of my PhD journey but one of the most crucial. Without your contribution to online optimisation, the thesis would be many pages shorter than it is now. Together we completed one of the most complex experiments in the OJF involving real-time shape optimisation of the morphing wing, with numerous algorithms, hardware and software running in unison. Your input was critical to its successful completion. Within your MSc, you managed to author two papers and co-author another paper on top of completing a complex wind tunnel experiment, which is truly exceptional.

Bo, thank you for your exceptional work on the incremental model-based control, which was the critical component for the backlash compensation and the entire experimental campaign following. Bo and Sherry, I think back with joy on the days and evenings we spent together in the OJF until security came to escort us home. I am honoured to be your 大哥-dàgē (the elder brother) and wind tunnel Captain.

Iren, you onboarded SmartX amid preparations for the project's most intense wind tunnel campaign. With very limited time, your experience troubleshooting a plethora of issues turned out to be a lifesaver for the whole campaign. More importantly, you owned the GVT effort and supported the experimental identification, two critical components of the campaign. Without your contributions, the major wind tunnel campaign would not be possible. Finally, your unreserved effort and unwavering support in brainstorming, proofreading and helping to improve this dissertation have been paramount to its completion.

From AS&CM, Laura, thank you for being the pillar of the AS&CM and fixing the important office stuff for us. Always attentive, positive and taking care of everything and everyone. "You're a star", as the British say.

Thanks to my office colleagues who made me feel very welcomed as a "freshman" PhD candidate and with whom we had many interesting discussions at the Friday beer or the coffee corner: Mario, Javier, Paul, Jassir, Marta, Ines, Tito, Natalia, Jens, Edgar, Eddy, Kristofer, Luc, Andres, Hammad. Mario, you impressed me from the beginning when we met; knowledgeable, in control, cool and always classy. Thanks for showing me around. Paul, I enjoyed many discussions we had, and people rightfully called you a walking encyclopedia on general aviation. Not only the latter but also being a NASTRAN-guru, thank you for helping me to get started with the wing model for the first FSI runs. Also, thank you (and Jurij) for your helping hand with the GG in the wind tunnel on many occasions. Javier, although our technical topics were far apart, we had many common interests and infinite topics for discussion. I could never hope to match your guitar skills. Still, I am very honoured you accepted the invitation to play at the Etsy fair in Rotterdam, which I co-organised with Audrey. Erik, thank you for the bike rides and for keeping up the "grandma-pace" on your 1000cc Street Triple so that I could keep up with you on my 500ccs.

Lastly, every great journey needs an even greater fellowship to share the burden. Mam djan, Sis djan, without your unwavering support, I would not be the person I am today. You both were there when all else failed and at times when I felt there was no outcome. Mam djan, you are the Galadriel in the fellowship, not just for my PhD, shining infinite light and wisdom on my path even where there was no path to be seen. Thank your unconditional love, encouragement and patience. Sis djan, I have met many bright people throughout my journey, but none are as brilliant as you. Able to arrive at the most rigorous, elegant solution, I don't think there is a problem in this world that you cannot solve. You were both Gandalf, always wise and strong, and Legolas, battling alongside with unwavering support. At times, you could even rival the great Smaug, if not to return what I rightfully deserved, but also to entertain little Lévon. Dear Lévon, you are still a little too young but well on your way to becoming a young and proud Legolas. Marc, I will spare you a silly Tolkien name. Thanks for your support in many ways to "the crazy fellowship" with knowledge, reasoning and calmness of Slash. Pap djan, you got me into engineering and helped me develop my problem-solving skills. There is no engineering challenge you cannot solve; with a little bit of Led Zeppelin in the background. I learned so much from you. Lévon, Marc, Tati, Papi, Hopar and all my family. And finally, Vosje. You came into my life in the midst of the chaos – non-stop experiment, sleepiness nights, paper deadlines – but it feels like you have been there forever. You are my Arwen, always supporting me with love and care without a reserve. Thank you, my dearest.

To all of you – you all share the credit for this achievement.

Tigran Mkhoyan
Lucerne, September 2022

Curriculum Vitæ

Tigran Mkhoyan

Tigran Mkhoyan was born in Yerevan, Armenia, and grew up in the city of Erdenet, Mongolia. After moving to the Netherlands, he attended the Johan De Wit Gymnasium in Dordrecht, where he obtained his grammar school diploma with the language and profile "Latin, Science, and Biology".

In 2015 Tigran received his BSc from the Faculty of Aerospace Engineering at Delft University Technology. He later continued his MSc with the Delft University of Technology, Department of Control and Operations, specialising in advanced control and motion cueing for Dynamic Flight Simulation. In 2017, he completed his master's thesis research entitled, *"Minimisation of Coriolis Effects in Human Centrifuges"* in collaboration with the Desdemona B.V., where he developed a motion cueing filter for the Desdemona Motion Simulator, helping pilots experience less motion sickness arising from the Coriolis effect. For his internship project at Desdemona/MultiSim B.V., he developed the model, software, and hardware of an F-16 flight simulator from a genuine F-16 cockpit for pilot training.

In late 2017 he started his PhD at the Faculty of Aerospace Engineering of the Delft University of Technology, Department of Aerospace Structures and Materials, working on the Smart-X project in collaboration with two other PhDs from Novel Aerospace Materials (NovAm) and Non-Destructive Testing (NDT) groups. For this project, he architected the autonomous smart morphing wing, the SmartX-Alpha. and with the SmartX team was awarded the best hardware prize in AMSE's Premier Conference on Smart Materials, Adaptive Structures, and Intelligent Systems conference in Irvine, CA, in 2020. His submitted research paper at this conference entitled "Design of a Smart Morphing Wing Using Integrated and Distributed Trailing Edge Camber Morphing" was also the runner-up for the best paper award. The SmartX-Alpha was later nominated as one of the 20 best Dutch tech ideas of 2021 and featured in the National Geographic article on green aviation: 'How green, how soon?'.

In 2018 Tigran took part in the summer school at EIT Digital in Stockholm, working on the Swift Wind headset device's conceptual design, intended to increase drivers' awareness through analysis of brain wave patterns. He won the best presentation award with his team at Ericsson headquarters in Kista, Sweden, in 2018. In 2019 he received a travel grant from IEEE RAS and attended the international summer school IEEE RAS Deep Learning for Robot Vision in Chile, Santiago. In 2019 he was invited for a keynote presentation in session Deep Insight, Frontiers of AI, at the Global Innovation Forum in Yerevan, Armenia, alongside Google Deepmind.

Being an aviation enthusiast, Tigran was part of the flight-coordinators team at Control & Simulation department, coordinating the yearly in-flight student practical on-board of the faculty's flying lab, Cessna Citation II (PH-LAB). The practical was intended to gather measurements of aircraft characteristics and eigenmotions, with a closing demonstration being the zero-g parabolic flight. At the end of his PhD, Tigran had accumulated over 20 flight hours and had become highly comfortable with zero-g parabolic flight.

Tigran also has a passion for human space flight, aside from aviation, which prompted him to continue the research he had started during his master's and investigate the Coriolis Effect in

Rotating Space Platforms for human space travel. In 2019 Tigran presented this work at the ASIC 2019 conference in Darmstadt in the Artificial Gravity session, alongside inspiring names in this field such as Prof. Laurence R. Young. This work was later followed by a publication in the ROOM space journal.

During his research, he led a series of wind tunnel test campaigns in collaboration with other researchers working on multidisciplinary topics and was involved in various dissemination activities. In 2020 Tigran was invited for keynote presentations at the SAMPE 2020 Europe conference, entitled "The SmartX Project". In 2022 he was invited for a keynote lecture at the Netherlands Association of Aeronautical Engineers in NLR headquarters in Amsterdam.

In the final year of his PhD, he independently developed the second hardware demonstrator SmartX-Neo. In 2022 he presented the paper entitled *"Aeroelastic Wing Demonstrator with a Distributed and Decentralized Control Architecture"* at the American Institute of Aeronautics and Astronautics (AIAA) SciTech Conference in San Diego, CA. For this research, he received the highest ranked Aerospace Design and Structures AIAA SciTech student paper award, the Jefferson Goblet.

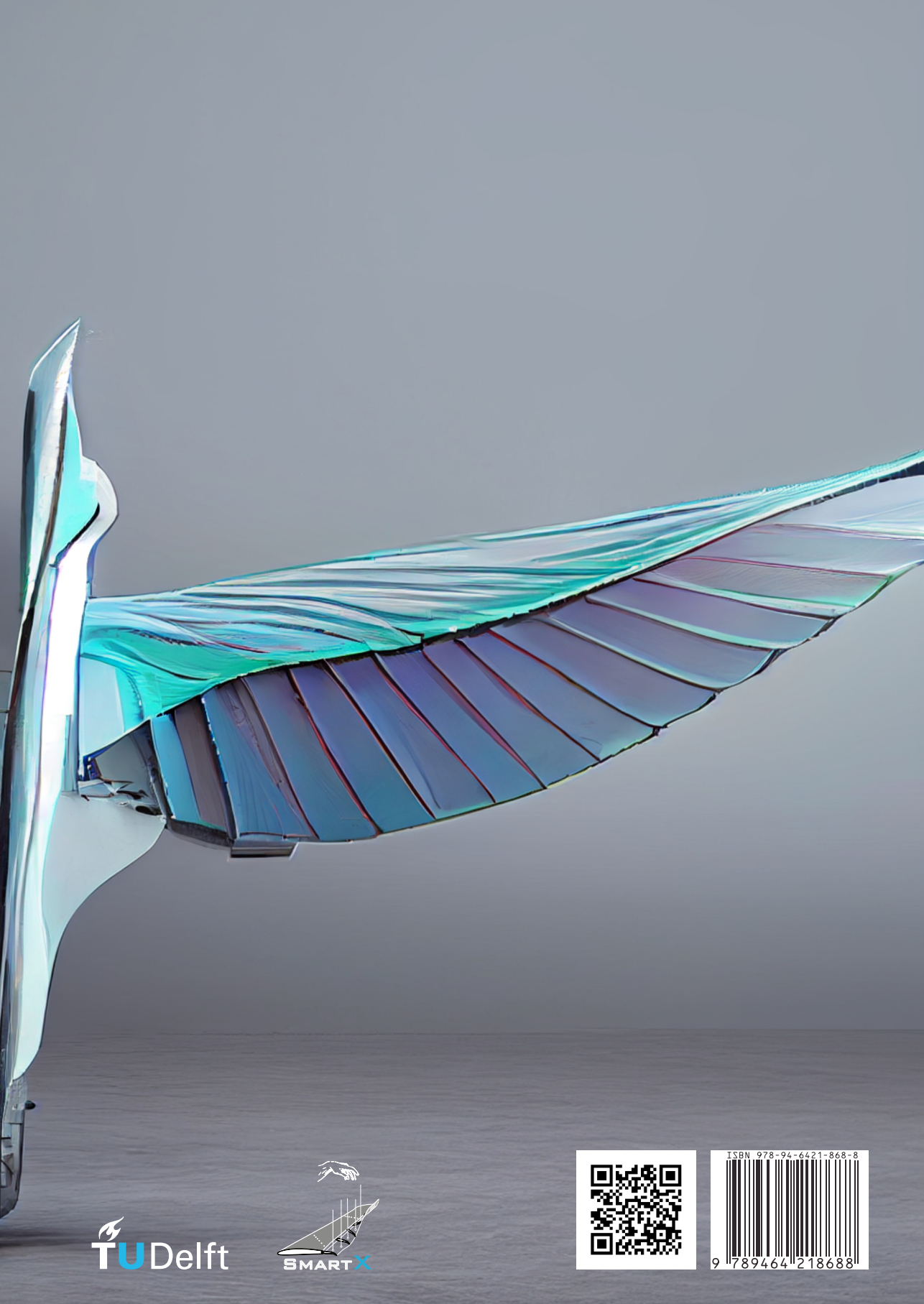
Tigran is determined to bridge the gap between aeroelasticity, design and advanced control and push the advancements in aircraft technology toward a new generation of energy-efficient, advanced aircraft concepts. After his defence in September 2022, he will continue his career at Aurora Flight Sciences, a Boeing Company, located at the Aurora Swiss Aerospace office in Lucerne, Switzerland.

AWARDS AND NOMINATIONS

2022 (Jan 05): Winner *Jefferson Goblet* at AIAA SciTech 2022 Conference, San Diego, CA, USA, paper:
"Aeroelastic Wing Demonstrator with a Distributed and Decentralized Control Architecture".
2021 (Sep 14): Nominee "Best Tech Idea" science magazine KIJK, Amsterdam, the Netherlands.
2020 (Sep 15): Winner *Best Student Hardware* ASME SMASIS 2020 Conference, Irwine, CA, USA
2020 (Sep 15): Runner-up *Best Student Paper* ASME SMASIS 2020 Conference, Irwine, CA, USA:
"Design of a Smart Morphing Wing Using Integrated and Distributed Trailing Edge Camber Morphing".
Systems (SMASIS) Conference: Seamless Smart Morphing wing, SmartX-Alpha. Short video.
2020 (Jun 02) Winner contest modelling of the virus spread, "COVID-19 spread model". Award by the President of the Republic of Armenia, Armen Sarkissian, Virtual.
2018 (May 25). Pick of the Week MATLAB for submission: simulinkParser, *"Manage Simulink Data Variations with simulinkParser"*, Blog Mathworks.

PRESS COVERAGE

2022 Graauw, S. (Sep 1, t.b. aired). Dutch TV *KRO-NCRV, NPO 1*, documentary by Andre Kuipers, "Ruimteschip Aarde, Ep. : Goldsmithing, Art & Morphing wings".
2022 Hattum, R. (Jun 20). Dutch TV NPO2, doc. by VPRO Tegenlicht, "Mobiliteit van morgen".
2021 Ouddeken, M. (July 1). Dutch TV NPO2, reportage "Het nieuwe vliegen: De kennis van NU"..
2021 Verhovek, H.S. (Sep 24). "How green, how soon?". *National Geographic*.
2020 Schenk, D. (Dec 11). " Duurzame vliegtuigen zijn radicaal anders". *NRC*.
2020 Kauffman, J. (Dec). "On-orbit and wind tunnel tests ... potential of morphing structures". Aerospace America.
2020 Wassink, J. (Okt 6). "Flexible wing flies with feeling". *TU Delta*.
2020 Mkhoyan, T. (Mar 31). "COVID-19: modelling the spread dynamics of the novel coronavirus". Linkedin.



TU Delft

SMART X

ISBN 978-94-6421-868-8

ISBN 978-94-6421-868-8

SMARTX

TU Delft