The Architecture of CubeSats and PocketQubes

Towards a Lean and Reliable Implementation of Subsystems and Their Interfaces

Bouwmeester, J.

DOI
10.4233/uuid:acac18a9-0dc1-46dc-8abb-045140a07b51

Publication date
2021

Document Version
Final published version

Citation (APA)
Bouwmeester, J. (2021). *The Architecture of CubeSats and PocketQubes: Towards a Lean and Reliable Implementation of Subsystems and Their Interfaces*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:acac18a9-0dc1-46dc-8abb-045140a07b51

The Architecture of CubeSats and PocketQubes

Towards a Lean and Reliable Implementation of Subsystems and Their Interfaces

The Architecture of CubeSats and PocketQubes

Towards a Lean and Reliable Implementation of Subsystems and Their Interfaces

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof.dr.ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Monday 20, September 2021 at 12:30 o'clock

by

Jasper BOUWMEESTER

Master of Science in Aerospace Engineering,
Delft University of Technology, the Netherlands
born in 's-Gravenzande, the Netherlands

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof.dr. E.K.A. Gill,	Delft University of Technology, promotor
Dr. A. Menicucci,	Delft University of Technology, copromotor
Independent members:	
Prof.dr. P.H.A.J.M. van Gelder,	Delft University of Technology
Prof.dr. G.C.H.E de Croon,	Delft University of Technology
Prof.dr. U.H. Walter,	Technical University of Munich, Germany
Prof.dr. J. Eickhoff,	University of Stuttgart, Germany
Prof.dr. T. Vladimirova,	University of Leicester, United Kingdom

Keywords: CubeSat, PocketQube, interface, architecture, satellite subsystem, reliability, redundancy, testing

Printed by: Ipskamp Printing

Front & Back: An artist impression of Delfi-n3Xt with a small photo cut-out of Delfi-C^3 on the back.

ISBN 978-94-6366-432-5

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

Try to learn something about everything
and everything about something.

Thomas H. Huxley

Contents

Summary

This thesis provides an innovative architecture for CubeSats and PocketQubes to improve their performance and reliability. CubeSats and PocketQubes are standard satellite form factors composed of one or more cubic units of 10 cm and 5 cm respectively. It is found that the current modular subsystem approach and the electrical interfaces are not optimal in terms of performance and use of technical resources. Reliability is also a concern for CubeSats. Only 35% are able to achieve full mission success. Available literature provides no comprehensive studies on these matters and there is thus a gap in knowledge to be filled. The objective of this study is to identify and quantify the performance and reliability issues related to the physical arrangement of subsystems and the electrical interfaces and to develop an innovative bus architecture which is reliable, flexible and allows for increased performance. The overarching research question is: *"Which satellite bus architecture provides a reliable solution to the needs and constraints of a CubeSat and a PocketQube mission?"*

In the first step, the electrical bus interfaces have been investigated. The key electrical interfaces are the data and power bus. The implementation and reliability of electrical bus interfaces has been analysed through a literature survey and a questionnaire. As a result, at least 65% of the 60 CubeSats in this survey do not to fulfill all mission objectives. While there is no evidence that the electrical interfaces are a major cause for this, there are still significant issues identified. Currently, the most popular data bus for CubeSats is I^2C. This data bus shows many bus lockup issues on CubeSats as well as a few catastrophic failures. The majority of failures in the electrical power subsystem are not allocated to the power bus interface. However, of the five CubeSats analysed which have implemented power distribution lines without failure protection, two have failed after a few days in orbit. Concerning the electrical power subsystem architecture, it is observed that different suppliers use different power distribution implementations in terms of supply lines and voltages. Power losses through stacked connectors, or alternative wiring, are limited and should not be a driving factor in trade-off analyses. The number of voltage level conversion steps in series can, however, best be limited for maximum efficiency. For the majority of the CubeSats which implemented the PC/104 connector, it is reported that the connector is too large. However, no catastrophic in-orbit failures have been reported because of this connector. Based on these findings, a need for a lean and robust electrical interface is identified.

Following the investigation on electrical power interfaces, an extensive trade-off has been performed to find an optimal solution to the identified issues and needs. The proposed interface standard comprises a linear data bus which is used for housekeeping data, internal commands and small-to-moderate payload data. A community-based analytic hierarchy process is used for the trade-off of design

options, resulting in the selection of RS-485 as standard data bus, mainly due to its low power consumption and high effective data throughput compared to other candidates. Concerning the electrical power interfaces, several switched and protected battery voltage lines are connecting the central electrical power subsystem unit to the other subsystems to enable a simple and efficient power distribution. The harness comprises a 14 and 9 pin stack-able connector for CubeSats and PocketQubes, respectively. This has led to the public release on the new interface standards CS14 and PQ9. The CS14 connector has only 8% of the footprint of the PC/104 connector. Moreover, the new interface offers higher data rate and lowers electrical losses compared to the options of PC/104 and the risk for compatibility issues is decreased.

Next, the high level architecture has been investigated in terms of the physical breakdown and allocation of subsystems and its components. The dominant existing architectural approach in the design of CubeSats and PocketQubes is the use of modular physical units, each hosting (parts of) subsystems. Some satellites also host subsystems or experimental payloads with an alternative approach, e.g. with cellularization of components or the integration of functions from different virtual subsystems into a single physical unit. These innovative concepts also have been investigated and proposed in other studies for implementation at the entire satellite. Cellularization of complete satellite segments and a satellite which comprises only of pyramid-shaped panels are examples of this. While they offer promising advantages when implemented smartly as part of a new architecture, their disadvantages become dominant when such a concept is implemented for the entire satellite. A smartly chosen hybrid of several concepts is investigated instead. Advanced flat panels on the outside of the satellite mixes the cellularized and panel concept. They integrate many components which interact with the environment with the aim to optimize volume and simplify satellite integration. Internally, modular systems are still used, but several classical core subsystems can be integrated towards a single core unit. Together with the lean electrical interface standard, the available payload volume is increased to 76% of the total satellite volume for a case study on the Delfi-n3Xt 3U CubeSat. This is a major improvement compared to the launch configuration, which has only 8% of the satellite volume available for the payload due to a classical modular approach and implementation of redundant subsystems.

In the final step, the reliability of CubeSats has been investigated. The lean electrical interface and the proposed advanced architecture mainly focus on improved performance, easier integration and volume reduction of the satellite bus. A few reliability issues have been mitigated, but overall reliability of CubeSats over its life time is still a concern. The implementation of redundant subsystems is a common method to improve reliability. This would, however, decrease payload volume again. Furthermore, it is not fully compatible with the proposed advanced architecture and the proposed electrical interfaces do not offer intrinsic redundancy. A common method to improve reliability in terms of development is to expand and improve the test campaign. Therefore, it is investigated which approach leads to more reliable CubeSats: applying subsystem redundancy or improvement of testing. A questionnaire has been used to collect data on the reliability of satellites

and subsystems. A variety of suitable failure distribution models is defined and the maximum likelihood of these models are estimated with the obtained failure data. The results are compared on several criteria. A product of a Lognormal distribution and a Gompertz distribution, addressing immaturity failure and wear-out respectively, is found to be the best representation of CubeSat reliability. Because CubeSat survival and failure data become sparse after a few years in-orbit, it is difficult to provide a sound estimate of the wear-out parameters using a maximum likelihood estimator. Therefore, Bayesian inference was applied using the estimate of existing failure data on small satellites as prior for the CubeSat estimate. This satellite level estimate was consequently used as prior for estimating the individual subsystem failure distributions. A reliability model for CubeSats with redundant subsystems is established and applied in a Monte Carlo simulation. Instead of allocating project resources to implement subsystem redundancy, these resources can also be allocated to more intensive testing of non-redundant subsystems to reduce the risks of immaturity failure. This is also modelled and compared to the results for CubeSats with redundant subsystems. The simulation results show that allocating project resources to improved testing is superior to allocating them to subsystem redundancy.

It has been found that CubeSats, and in their wake also PocketQubes, can have an improved architecture compared to a highly modular architecture. Compared to existing standards, the proposed lean electrical interfaces CS14 and PQ9 are smaller, yield lower power losses, allow for higher data rates and reduce the risk of compatibility issues between subsystems. A smarter physical architecture integrates parts of the satellite hardware of the multiple subsystems which can be found in most satellites into a single physical core unit. The electronic systems which are directly related to sensors or antennas can be moved to advanced outer panels. Mission specific components, such as propulsion and attitude actuators, can best still be kept modular. This innovative architecture, which is a hybrid of advanced concepts and the modular architecture, allows for easier integration and increased relative payload volume. Improvement of reliability should mainly be tackled by extending and intensifying test campaigns before redundancy is being considered. The failure classification, reliability models, estimation methods and reliability simulation models in this research are considered to be useful for other satellite classes and complex systems as well.

Samenvatting

Dit proefschrift biedt een innovatieve architectuur voor CubeSats en PocketQubes om de prestaties en betrouwbaarheid ervan te verbeteren. CubeSats en PocketQubes zijn standaarden voor satellieten van een of meer kubieke eenheden van respectievelijk 10 cm en 5 cm. Het blijkt dat een modulaire subsysteem-benadering en de elektrische interfaces niet optimaal zijn in termen van prestaties en gebruik van technische middelen. Betrouwbaarheid is ook een zorg voor CubeSats. Slechts 35% is in staat om het volledige missie-succes te behalen. De beschikbare literatuur biedt geen alomvattende studies over deze zaken en er is dus een gebrek aan kennis die moet worden opgevuld. Het doel van deze studie is om de prestatie- en betrouwbaarheidsproblemen met betrekking tot de fysieke toewijzing van subsystemen en de elektrische interfaces te identificeren en kwantificeren en om een innovatieve architectuur te ontwikkelen die betrouwbaar en flexibel is en betere prestaties mogelijk maakt. De overkoepelende onderzoeksvraag is: *"Welke satelliet-bus-architectuur biedt een betrouwbare oplossing voor de behoeften en beperkingen van een CubeSat- en een PocketQube-missie?"*

In de eerste stap zijn de elektrische bus-interfaces onderzocht. De belangrijkste elektrische interfaces zijn de data- en vermogensbus. De implementatie en betrouwbaarheid van elektrische bus-interfaces wordt geanalyseerd door middel van een literatuuronderzoek en een vragenlijst. Als resultaat hiervan blijkt ten minste 65% van de 60 CubeSats in deze enquête niet aan alle missiedoelen te voldoen. Hoewel er geen aanwijzingen zijn dat de elektrische interfaces hier een belangrijke oorzaak van zijn, zijn er nog steeds belangrijke problemen geïdentificeerd. Momenteel is de meest populaire databus voor CubeSats I^2C. Deze databus toont veel vastloop-problemen op CubeSats, evenals een paar catastrofale storingen. De meeste storingen in het subsysteem voor elektrisch voeding worden niet toegewezen aan de elektrische interfaces. Van de vijf geanalyseerde CubeSats die distributielijnen hebben geïmplementeerd zonder uitvalbeveiliging, zijn er echter twee mislukt na een paar dagen in een baan om de aarde. Wat betreft de architectuur van het subsysteem voor elektrisch vermogen, wordt opgemerkt dat verschillende leveranciers verschillende implementaties van distributie gebruiken in termen van voedingslijnen en spanningen. Verlies van vermogen door gestapelde connectoren of alternatieve bedrading zijn beperkt en mogen geen drijvende factor zijn bij het afwegen van zaken. Het aantal seriële conversiestappen van het spanningsniveau kan echter het beste worden beperkt voor maximale efficiëntie. Voor de meerderheid van de CubeSats die de PC/104-connector hebben geïmplementeerd, wordt gemeld dat de connector te groot is. Vanwege deze connector zijn er echter geen catastrofale storingen in de omloopbaan gemeld. Op basis van deze bevindingen wordt vastgesteld dat er behoefte is aan een slanke en robuuste elektrische interface.

In navolging van het onderzoek naar elektrische bus-interfaces, is er een uitgebreide afweging gemaakt om een optimale oplossing te vinden voor de geïdentificeerde problemen en behoeften. De voorgestelde interface-standaard omvat een lineaire databus die wordt gebruikt voor huishoudgegevens, interne commando's en gegevens van kleine tot gemiddelde payloads. Een analytisch hiërarchieproces wordt gebruikt voor de afweging van ontwerpopties, waarbij ook de 'space community' wordt betrokken. Dit resulteert in de selectie van RS-485 als standaard databus, voornamelijk vanwege het lage stroomverbruik en de hoge effectieve gegevensdoorvoer in vergelijking met andere opties. Voor de elektrische stroom interfaces worden verschillende geschakelde en beschermde voedingslijnen gebruikt om het centrale elektrische vermogens-subsysteem met de andere subsystemen te verbinden om zodanig een eenvoudige en efficiënte vermogensdistributie mogelijk te maken. De bedrading bevat een 14- en 9-pins stapelbare connector voor respectievelijk CubeSats en PocketQubes. Dit heeft geleid tot de publicatie van de nieuwe interface-standaarden CS14 en PQ9. De CS14-connector heeft slechts 8 % van de afdruk van de PC/104-connector. Bovendien biedt de nieuwe interface een hogere datasnelheid en lagere elektrische verliezen in vergelijking met de opties van PC/104 en wordt het risico op compatibiliteitsproblemen verkleind.

In de volgende stap wordt de architectuur op hoog niveau onderzocht in termen van de fysieke uitsplitsing en toewijzing van subsystemen en zijn componenten. De huidige dominante benadering van het ontwerp van CubeSats en PocketQubes is het gebruik van modulaire fysieke eenheden, die elk (delen van) subsystemen bevatten. Sommige satellieten bevatten ook subsystemen of experimentele ladingen met een alternatieve benadering, b.v. met cellularisatie van componenten of de integratie van functies van verschillende virtuele subsystemen in een fysieke eenheid. Deze innovatieve concepten zijn ook onderzocht en voorgesteld in andere studies voor implementatie op de gehele satelliet. Cellularisatie van complete satellietsegmenten en een satelliet die alleen uit piramidevormige panelen bestaat, zijn hier voorbeelden van. Hoewel ze veelbelovende voordelen bieden wanneer ze slim worden geïmplementeerd als onderdeel van een nieuwe architectuur, worden hun nadelen dominant wanneer een dergelijk concept wordt geïmplementeerd voor de hele satelliet. In plaats daarvan wordt een slim gekozen hybride van meerdere concepten onderzocht. Geavanceerde platte panelen aan de buitenkant van de satelliet combineren het cellulaire en paneel-concept. Het integreert vele componenten die interacteren met de omgeving met als doel het volume te optimaliseren en de satellietintegratie te vereenvoudigen. Intern worden nog steeds modulaire systemen gebruikt, maar verschillende klassieke kernsubsystemen kunnen worden geïntegreerd tot een enkele kerneenheid. Samen met de slanke elektrische interface-standaard wordt het beschikbare laadvolume verhoogd tot 76% voor een Delfi-n3Xt 3U CubeSat casus. Dit is een grote verbetering ten opzichte van de lanceer-configuratie, die slechts 8% van het laadvolume oplevert vanwege de klassieke modulaire benadering en de implementatie van redundante subsystemen.

In de laatste stap wordt de betrouwbaarheid van CubeSats onderzocht. De slanke elektrische interface en de voorgestelde geavanceerde architectuur zijn voornamelijk gericht op verbeterde prestaties, eenvoudigere integratie en volumever-

mindering van de satelliet-bus. Een paar betrouwbaarheidsproblemen zijn verholpen, maar de algehele betrouwbaarheid van CubeSats gedurende zijn levensduur is nog steeds een punt van zorg. De implementatie van redundante subsystemen is een veelgebruikte methode om de betrouwbaarheid te verbeteren. Dit zou echter het payload-volume weer verkleinen, is niet volledig compatibel met de voorgestelde geavanceerde architectuur en de voorgestelde elektrische interfaces bieden geen intrinsieke redundantie. Een veelgebruikte methode om de betrouwbaarheid in termen van ontwikkeling te verbeteren, is het uitbreiden en verbeteren van de testcampagne. Daarom wordt onderzocht welke aanpak leidt tot betrouwbaardere CubeSats: het toepassen van redundantie van subsystemen of het verbeteren van testen. Met behulp van een vragenlijst zijn gegevens verzameld over de betrouwbaarheid van satellieten en subsystemen. Er is een verscheidenheid aan potentieel geschikte kans-distributie-modellen gedefinieerd en de maximale waarschijnlijkheid van deze modellen wordt geschat met de verkregen gegevens. De resultaten worden op verschillende criteria vergeleken. Een product van een lognormale distributie en een Gompertz-distributie, die respectievelijk falen door onvolwassenheid en falen door slijtage adresseren, blijkt de beste weergave van de CubeSat-betrouwbaarheid te zijn. Omdat de overlevings- en faalgegevens van CubeSats schaars worden na een paar jaar in een baan om de aarde, is het moeilijk om een goede schatting te maken van de slijtageparameters met behulp van de maximale-waarschijnlijkheids-schatter. Daarom wordt Bayesiaanse inferentie toegepast met behulp van de schatting van bestaande faalgegevens op kleine satellieten als a-priori-kansverdeling voor de CubeSat-schatting. Deze schatting op satellietniveau wordt vervolgens gebruikt als a-priori voor het schatten van de individuele kansverdelingen van elk subsysteem. Een betrouwbaarheidsmodel voor CubeSats met redundante subsystemen wordt opgesteld en toegepast in een Monte Carlo-simulatie. In plaats van projectmiddelen toe te wijzen om redundantie van subsystemen te implementeren, kunnen deze middelen ook worden toegewezen aan intensievere tests van niet-redundante subsystemen om het risico van falen door onvolwassenheid te verminderen. Dit wordt ook gemodelleerd en vergeleken met de resultaten voor CubeSats met redundante subsystemen. De simulatieresultaten laten zien dat het toewijzen van projectmiddelen aan verbeterde tests substantieel meer effect heeft dan het toewijzen ervan aan subsysteemredundantie.

Het is gebleken dat CubeSats, en in hun kielzog ook PocketQubes, een verbeterde architectuur kunnen hebben in vergelijking met een sterk modulair ontwerp. In vergelijking met bestaande standaarden zijn de voorgestelde slanke elektrische interfaces CS14 en PQ9 kleiner, leveren ze minder vermogensverliezen op, maken ze hogere datasnelheden mogelijk en verminderen ze het risico op compatibiliteitsproblemen tussen subsystemen. Een slimmere fysieke architectuur integreert delen van de satelliet-hardware van de meerdere subsystemen, die in de meeste satellieten te vinden zijn, in een enkele fysieke kerneenheid. De elektronica die direct gerelateerd is aan sensoren of antennes kan worden verplaatst naar geavanceerde platte panelen aan de buitenzijde. Missie-specifieke componenten, zoals voortstuwings- en stand-regeling-actuatoren, kunnen nog het beste modulair worden gehouden. Deze innovatieve architectuur, die een hybride is van geavanceerde

concepten en de modulaire architectuur, zorgt voor eenvoudigere integratie en een groter relatief payload-volume. Verbetering van de betrouwbaarheid dient vooral te worden aangepakt door testcampagnes uit te breiden en te intensiveren voordat redundantie wordt overwogen. De foutclassificatie, betrouwbaarheidsmodellen, schattingsmethoden en betrouwbaarheidssimulatiemodellen in deze studie worden ook als bruikbaar beschouwd voor andere satellietklassen en complexe systemen.

Preface

I performed my PhD research next to a job as satellite project manager and lecturer. The experience and insights I gained during my work have been a key motivation to start a PhD topic relevant to my work and they helped me to work efficiently. However, the work on education and the projects can be demanding. They often gain higher priority than the PhD research when people are dependent on you. My PhD research therefore has shifted gear often. The final phase of this research has been performed during the COVID-19 pandemic period. Despite its depressive effects it eventually motivated me to perform a final marathon towards completion, utilizing the lack of travel, social interaction and other distractions. Completing a thesis in such isolation can be helpful, but without the professional and social interaction with other people I would never have been able to complete this thesis.

I would like to thank my promoter Eberhard Gill for his guidance on the research and his formal support to allow me to perform this research in part of the time on the job. His open questions during our meetings and detailed comments on my drafts aided the continuation and attainment of the appropriate level. I would like to thank my co-promoter Alessandra Meniccuci for her guidance. She has been very flexible in supporting the PhD research after joining mid-way, when most content-related decisions had already been made. She guided me during the reliability analysis and provided thorough reviews of all chapters.

I would like to thank Martin Langer for his collaboration on the extensive CubeSat survey and acting as sparring partner in our Skype conversations. Furthermore, I would like to thank Ales Povalac, Stefan van der Linden, Nuno Santos, Stefano Speretta and Sevket Uludag who where involved in part of the research and co-authored a related article. I would like to thank my colleague Hans Kuiper for providing the necessary perspective and humor on the job, Marco Kleuskens for having the courage to act as paranimf and all my other friends for asking about the status of my PhD on each possible occasion.

I would like to thank my parents who have supported and encouraged me to aim for the highest education possible. I would like to mention my sons Olivier and Serge. When I was stuck sometimes, they provided inspiration to continue my efforts by seeing them overcome their first hurdles at primary school. Finally, I would especially like to thank my wife Kim for her mental support and encouragement to never give up and continue my PhD research. During the COVID-19 lock-downs and school vacations she often took full care of the kids such that I could concentrate and accelerate my progress.

Jasper Bouwmeester
Dordrecht, July 2021

1

Introduction

Great things are done
by a series of small things
brought together.

Vincent van Gogh

What is now proved
was once only imagined.

William Blake

This study focuses on the system level architecture of CubeSats and PocketQubes. Statistical and qualitative analysis is performed in order to determine the impact of the architecture on the performance and reliability of these small satellite platforms and how these can be improved. In this chapter, the research is introduced by providing its background, context and objectives.

1.1. Emergence of CubeSats and PocketQubes

In the first decades of spaceflight, the access to space was limited to governmental agencies. Due to increased launch reliability and the advancement in electronics, access to space slowly expanded to industrial players. By the end of the 20th century, the advancement of micro-electronics had reached a level in which satellites with limited functionality could be fitted in a very small volume. The launch cost was not prohibitive anymore for potential new players with limited budgets, such as universities and small companies. However, the infrastructure to launch very small satellites from inexperienced new players was still missing until the CubeSat concept was introduced in 1999 by Puig-Suari and Twiggs [1] of the California

1

Polytechnic State University. A CubeSat is a satellite with standardized mechanical interfaces for a launch interface adaptor and comprises one or more units of 10 × 10 × 10 cm (1U). The availability of micro-controllers, which are system-on-chips requiring less than a Watt to operate, have been a key enabler to the development of these satellites. The standardized launch interface has created a new industry to supply CubeSat subsystem hardware with standardized mechanical and electrical interfaces. Subsequently, this has helped new players in the space community, such as universities and new companies, to quickly start the development of satellites at a relatively low cost. The containment of the satellites during launch in their respective CubeSat interface adaptors, limiting the risk of causing damage to other nearby satellites, has helped to accept CubeSats as secondary or tertiary payloads on-board launch vehicles. The emergence of CubeSat launch service providers has increased access to space. As a result, space has become an affordable and reachable domain for those who were unable to do so in the past. The growth of launched CubeSats is visualized in Figure 1.1, with almost 1400 CubeSats launched in total by the end of 2020. It should be noted that this number comprises 418 CubeSats from the Flock constellation for Earth Observation from the company Planet and 126 CubeSats from the Lemur constellation for maritime navigation and weather forecasting of the company Spire Global. These constellations have a large influence on the annual launched CubeSats from 2014 on-wards. The number of unique CubeSat missions seems to have leveled off since then.

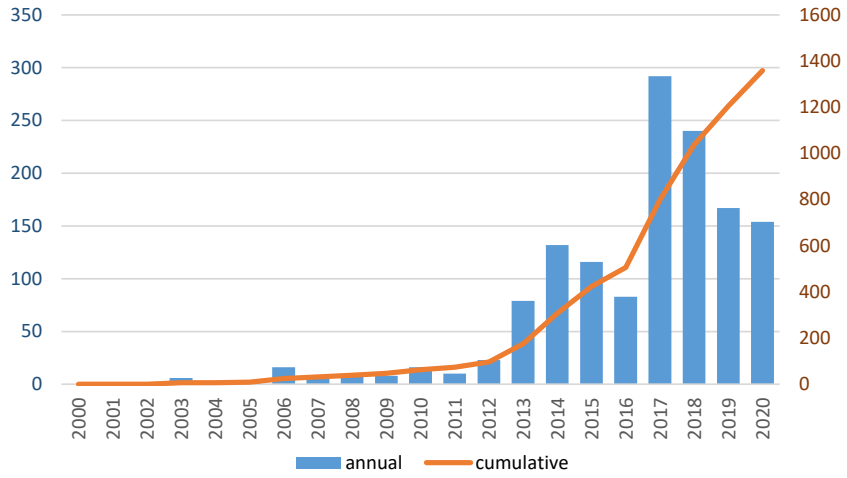


Figure 1.1: Number of CubeSats launched (inc. failed launches). Data retrieved from www.nanosats.eu (by E. Kulu) on 9-12-2020.

In 2006, the PCBSat concept is introduced by Barnhart, Vladimirova, Baker and Sweeting [2]. The philosophy is that the satellite is integrated on a single PCB. In terms of form factor, it is essentially a slice of a CubeSat of 0.2U [2] or 0.25U [3] and compatible with CubeSat launch deployment systems. A few years later an

alternative form factor was introduced for sub-kilogram satellites: the PocketQube. The PocketQube was first introduced by Twiggs in 2009 [4] in a cooperation between Morehead State University and GAUSS Srl. They developed a PocketQube deployment system 'MRFOD'. The general philosophy of the PocketQubes is similar to CubeSats, with the main difference that the cubic unit size is reduced to 5 cm, implying a volume reduction of a factor eight. The first batch of four PocketQubes has been launched in 2013. The PocketQube standard was updated and publicly released in 2018 [5] in a cooperation between TU Delft, Alba Orbital and GAUSS. In 2019, six PocketQubes have been launched according to this new standard. In the next four years, at least 40 more PocketQubes (of which many unique missions) are planned. Compared to CubeSats, PocketQubes are still in their infancy and growth in numbers is expected. Although PocketQube platforms provide more limited technical resources compared to CubeSats, they are excellent platforms to complement or even replace CubeSats for educational and training purposes [6]. Furthermore, niche applications are foreseen which require vast networks of satellites with relatively small power- and data-efficient payload. In this respect, they will complement rather than replace CubeSats [6].

TU Delft has its own series of very small satellites: the Delfi Program. This program comprises the Delfi-C^3 and Delfi-n3Xt CubeSats and the Delfi-PQ PocketQube. Delfi-C^3, shown in Figure 1.2, was launched in 2008. It has achieved full mission success and is still operational as of 2021. Delfi-n3Xt, shown in Figure 1.3, was launched in 2013. It has achieved primary mission success. Delfi-PQ, shown in Figure 1.4, is ready for a launch in 2021.

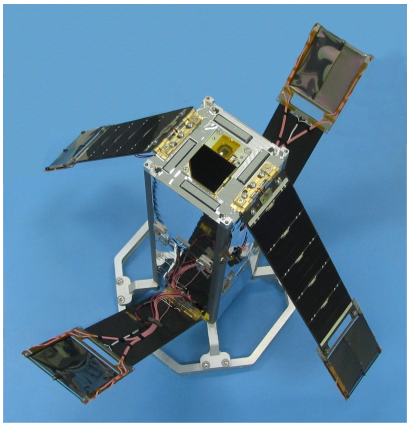


Figure 1.2: Delfi-C^3.

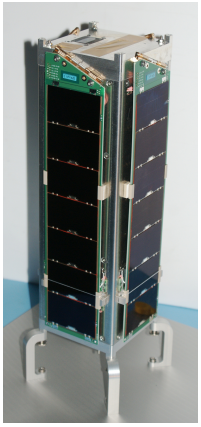


Figure 1.3: Delfi-n3Xt (stowed).

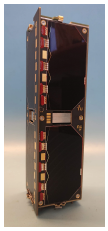


Figure 1.4: Delfi-PQ.

The author of this thesis has gained an extensive experience in this program as systems engineer on Delfi-C^3 followed by a role as project manager in the operational phase of Delfi-C^3, the end-to-end development of Delfi-n3Xt and the design phase of Delfi-PQ. The experience and insights gained over a decade on CubeSats has been a key motivation to start an intensive study in this field for a PhD thesis. During the study, PocketQubes were added where possible as they provide similar

1

characteristics to CubeSats: small standardized satellites using commercial-off-the-shelf components, opening access to space to new players.

1.2. Need for Satellite Bus Architecture Innovation

In a CubeSat survey performed in 2009 [7] it became evident that the failure rate of CubeSats was relatively high: more than half of the missions were not fully successful. From the 10 PocketQubes launched up to 2021, only 2 have been reported to be fully operational. During the development and operations of Delfi-C^3 and Delfi-n3Xt, technical problems were mainly associated with the interfaces and the methods for failure handling. Both are related to the satellite bus architecture. This has been a motivation to focus the study in this field.

The definition of the 'satellite bus' applied in this thesis is 'the combination of all satellite subsystems which facilitate the mission payload'. The definition of the 'bus architecture' as applied in this thesis is 'the functional and physical arrangement of subsystems, interfaces between subsystems and the operational strategy of distribution of data and power within the satellite'.

Industry and many CubeSat developers have adopted 'PC/104', which is a standard for the Printed Circuit Board (PCB) outline and a 104-pin stackable connector which offers many options for power distribution and data communication between the PCBs. Due to its high implementation rate it has become the de facto interface standard for CubeSats. In the bread-boarding phase it became clear that one of the biggest issues with the PC/104 standard is caused by its large volume. Therefore, for Delfi-C^3 and Delfi-n3Xt a custom electrical interface was developed. However, also for these interfaces issues are identified, such as low reliability and limiting data rate for the chosen data bus (I^2C) as well as a loss of power due to chosen electrical power distribution. These issues were already a design driver for this mission, so it was expected that it will limit performance in the future as subsystem capabilities increase. Both satellites were developed with a physical modularization of the subsystems. Due to a reliability philosophy to mitigate as many single-points-of-failure as possible for critical subsystems, redundancy was applied to many modular units. The modularization and subsystem redundancy has led to a relatively high volume for the satellite subsystems and limited volume for payloads. This was not an issue for these satellites since technology demonstration of some of the subsystems was one of the main mission objectives. This is however different for missions with applications requiring a relatively large payload.

After the successful launch and operations of Delfi-C^3 and Delfi-n3Xt, in 2016, TU Delft started the development of a PocketQube. Due to the even tighter technical budgets, the design of the electrical interfaces and the physical arrangement of the subsystems becomes yet more challenging for this type of small satellites.

Available literature provides no comprehensive studies on these matters. There is thus a need for an in-depth investigation which covers the interfaces as well as the physical arrangement of subsystems to create an innovative bus architecture which is reliable, flexible and allows for increased performance. Therefore, this PhD research aims to fill this gap in knowledge by analyzing the issues and limitations of the state-of-the-art and identifying potential solutions to improve the satellite bus.

1.3. Research Objective and Methodology

The objective of this PhD thesis is to answer the following overarching research question: *Which satellite bus architecture provides a reliable solution to the needs and constraints of a CubeSat and a PocketQube mission?* There are two sub-questions derived, which allow to divide this research into a part which investigates the problem into depth and a part which explores potential solutions to the problem:

1. What is the impact of the bus architecture on the reliability and performance of a CubeSat and a PocketQube?
   (a) What is the overall reliability of launched CubeSats? Which issues with the bus architecture can be identified and what is their relative impact on the overall reliability?
   (b) What is the state-of-the art performance of a CubeSat and PocketQube bus and which demand can be foreseen in the near future? What is the impact of the bus architecture on this performance?
   (c) Which reliability metrics and which estimation methods can best be used to model a bus architecture given the provided statistical information? What are the results?
2. Which innovative and reliable bus architecture meets the typical constraints and performance demands of CubeSats and PocketQubes foreseen in the near future?
   (a) Which options can be identified for an innovative CubeSat and PocketQube bus architecture which may tackle the reliability and/or performance issues of existing missions?
   (b) Which aspects related to satellite architecture and development mostly affect reliability? How do the options, related to these aspects, compare on reliability using an appropriate metric?

The general research method applied in this thesis is to investigate the issues and performance of the state-of-the art through extensive literature survey and data acquisition using questionnaires. This investigation is mainly based on CubeSats, because there is a multi-fold of literature and data available for these platforms compared to PocketQubes. However, even for CubeSats the number of data which could be obtained is still limited as not all teams respond to questionnaires or have published their results in sufficient detail. Next to this, due to the increase of launched CubeSats over time (see Figure 1.1), the long-term data on CubeSat failures is relatively limited. The challenge of sparse failure data will be tackled by identifying appropriate reliability models as well as statistical approaches. The key objective of this research is to identify, analyse and select innovative approaches to improve performance and reliability of CubeSats and PocketQubes. The exploration of innovative approaches will be based on own ideas and ideas found in literature. The analysis will be performed through modelling. Trade-offs will be based on clear criteria and analysis of results of tests and models. The CubeSat and PocketQube community will be involved for subjective parts of the trade-offs.

1.4. Thesis Outline

A CubeSat survey is presented in Chapter 2, which starts with general CubeSat statistics but focuses on the reliability and performance of electrical interfaces. The survey is based on literature and data acquisition through a questionnaire. In Chapter 3, a novel lean interface for CubeSats and PocketQubes is presented. The proposed interface is based on the findings of the survey of Chapter 2, a detailed technical analysis of existing interfaces, identification and tests of design options and a community based trade-off process. In Chapter 4, innovative architectures in terms of physical arrangement of subsystems and components are identified, analyzed and traded. The conclusions of Chapters 2, 3 and 4 have resulted in a crucial reliability question: whether reliability can best be improved by redundancy or by improvement of the test campaign. In Chapter 5 the CubeSat reliability is investigated, modelled and analyzed. Finally, in Chapter 6 conclusions are drawn from this research and an outlook for the future is provided.

References

[1] J. Puig-Suari, C. Turner, and W. Ahlgren, *Development of the Standard CubeSat Deployer and a CubeSat Class Picosatellite,* in *IEEE Aerospace Conference Proceedings* (2001).

[2] D. Barnhart, T. Vladimirova, A. Baker, and M. Sweeting, *A Low-cost Femtosatellite to Enable Distributed Space Systems,* Acta Astronautica **64**, 1123 (2009).

[3] R. L. Balthazor, M. G. McHarg, C. S. Godbold, D. J. Barnhart, and T. Vladimirova, *Distributed space-based ionospheric multiple plasma sensor networks,* in *IEEE Aerospace Conference Proceedings* (IEEE, 2009).

[4] R. Twiggs, *Making it Small,* in *Proceedings of the CubeSat Developers Workshop* (California Polytechnic State University, San Luis Obispo, 2009).

[5] S. Radu, M. S. Uludag, S. Speretta, J. Bouwmeester, E. Gill, and N. Foteinakis, *Delfi-PQ: the First PocketQube of Delft University of Technology,* in *Proceedings of the 69th International Astronautical Congress* (IAF, Bremen, 2018).

[6] J. Bouwmeester, S. Radu, M. S. Uludag, N. Chronas, S. Speretta, A. Menicucci, and E. K. Gill, *Utility and Constraints of PocketQubes,* CEAS Space Journal **12**, 573 (2020).

[7] J. Bouwmeester and J. Guo, *Survey of Worldwide Pico- and Nanosatellite Missions, Distributions and Subsystem Technology,* Acta Astronautica **67**, 854 (2010).

2

Survey on CubeSat Electrical Bus Interfaces

Latest surveys shows that 3 out of 4 people make up 75% of the world's population.

Stephen Hawking

This chapter provides results and conclusions derived from a survey on the implementation and reliability aspects of CubeSat bus interfaces, with an emphasis on the data bus and power distribution. It provides recommendations for a future CubeSat bus standard. The survey is based on a literature study and a questionnaire representing 60 launched CubeSats and 44 to be launched CubeSats. It is found that the bus interfaces are not the main driver for mission failures. However, it is concluded that the Inter Integrated Circuit (I^2C) data bus, as implemented in a great majority of the CubeSats, caused some catastrophic satellite failures and a vast amount of bus lockups. The power distribution may lead to catastrophic failures if the power lines are not protected against over-current. A connector and wiring standard widely implemented in CubeSats is based on the PC/104 standard. However, most participants find the 104 pin connector of this standard too large. For a future CubeSat bus interface standard, it is recommended to implement a reliable data bus, a power distribution with over-current protection and a wiring harness with smaller connectors compared with PC/104.

Parts of this chapter have been published in the CEAS Space Journal **2**, 9 (2017) [1]

2.1. Survey Motivation and Set-up

In the past, there have been several global surveys on CubeSats looking at their mission characteristics, implemented technologies and overall success rates [2, 3]. One study shows that about 40% of the first 100 CubeSats launched were not successful and provides some qualitative analysis on the failure causes with an emphasis on the development process [3]. It concludes with the recommendations to improve understanding of the failure sources and study technological capabilities to look for trends and prediction. The objective of this chapter is to provide the results from a new survey on the implementation and reliability of CubeSat data and power interfaces.

2.1.1. Motivation for the Survey

Standardization of interfaces has been one of the key factors of the increasing popularity of CubeSats in terms of missions [3] and commercially available subsystems. A standard for electrical interfaces allows for easy integration of subsystems from different suppliers. The CubeSat specification limits itself to external dimensions and interfaces with the launch adapter [4]. There is formally no standard specification for CubeSat electrical bus interfaces. However, the wiring harness of the PC/104 embedded systems standard [5] has been adopted in many CubeSats missions and commercially available CubeSat subsystems. Those using the PC/104 standard also widely implement the I^2C data bus for communication between subsystems. The PC/104 standard in combination with the I^2C data bus is currently being associated with the CubeSat standard. However, as will be elaborated in Section 2.5, the de-facto standard has been applied in a rather inconsistent manner leading to potential compatibility issues when integrating several commercially available subsystems. Next to this compatibility issue, many more problems with the I^2C data bus interface were found throughout the development and operations of Delfi-C^3 and Delfi-n3Xt [6], two successful CubeSats of TU Delft DelfiSpace program. A more extensive survey and investigation into the electrical bus interfaces is considered to be a good starting point for further investigation into reliability aspects of CubeSats, as such interfaces introduce requirements and constraints to all subsystems. Second, this survey is intended as a first step in a thorough analysis which can be used as input for the design of future CubeSats as well as the development of a new potential CubeSat standard which is reliable and fulfils the needs of future generations of CubeSats.

2.1.2. Survey Approach

First, a literature survey has been performed. In the majority of the publications on flight results, however, specific issues are not provided and/or discussed in detail. Therefore, the survey was complemented with an extensive questionnaire sent to the global CubeSat community. This provides a statistical basis, while the literature survey supports and substantiates some of the findings. Further literature study is subsequently used to analyse the results and to draw conclusions.

2.1.3. Conclusions from Other Satellite Surveys

The reliability of launched satellites has been investigated by several researchers worldwide. According to a study of 129 satellites of all classes [7], at least 45% of all satellite failures can be allocated to electrical faults. The electrical power subsystem (EPS) is accountable for 27% and Command and Data Handling (CDH) for 15% of the failures encountered, which together have a major impact on reliability. Looking more closely to the electrical interfaces, only 5% of the failures are allocated to the power bus and there is no specific mention of failures allocated to the data bus. Another study [8] focused on the EPS specifically using data of over a thousand spacecraft launched between 1990 and 2008. In Low Earth Orbit (LEO), which is the orbital regime of the vast majority of CubeSats, 29% of the failures observed in the electrical power subsystem are allocated to the electrical distribution. Of these failures, 80% are fatal ones. The overall average failure rate of the EPS is, however, just 3.8% for LEO satellites. These two studies do not confirm the substantial relevance of the electrical interfaces on the reliability of satellites, in general.

There has been a study on small satellite reliability which analyses the anomalies of subsystems of 222 satellites up to 500 kg [9]. This study provided reliability over operational lifetime. One of the conclusions is that satellites below 10 kg show a relatively high infant mortality rate and short lifetime compared with satellites between 10 kg and 500 kg. Telemetry, tracking and command (TT&C), the Thermal Control System (TCS), and the mechanisms and structures (M&S) contribute most to infant mortality, while the EPS contributes to the largest number of failures overall [9].

In a statistical study on the first 100 launched CubeSats performed in 2013 [3], mission failures of CubeSats are analysed on a high level. One major conclusion is that for a third of all failed missions radio signals have never been received after launch. Another 27% of the failures can be attributed to a configuration or interface failure between communication hardware and 14% to the EPS [3]. This study does not specifically address the electrical interfaces.

From all past satellite surveys, it can be concluded that electrical interfaces have not been identified as a significant contributor to mission failures. However, as CubeSats are just a small and relatively recent subset of the small satellite surveys [7–9] and the CubeSat survey [3] only shows reliability figures on a high system level, initial concerns stated in Section 2.1.1 cannot be relieved without further study.

2.2. General Survey Statistics

First, the questionnaire is introduced in Section 2.2.1. Section 2.2.2 provides overall CubeSat reliability results used to investigate the relevance of the electrical interfaces in particular.

2.2.1. Questionnaire Response and Processing

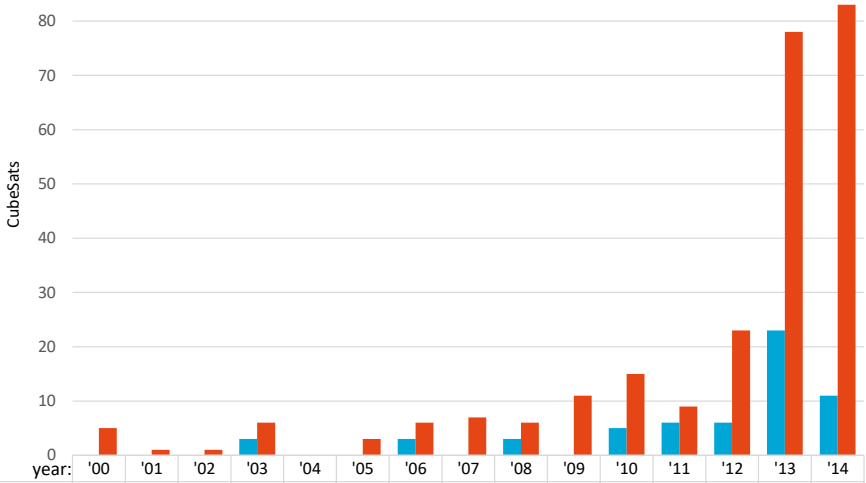
On the 4th of November 2014, a questionnaire has been sent out to in total 987 personal and general email-addresses affiliated with all of the launched CubeSats and many CubeSats in development at that time. It has been decided and communicated that the results of the questionnaire are treated anonymously, as full public disclosure might otherwise prohibit or discourage participation for some. The anonymous database can be found in the 4TU repository [10]. The questionnaire consisted roughly of three sections with a total of 33 questions. The first section addresses the general reliability aspects of the represented CubeSat, the second section addresses the reliability aspects of the bus interfaces of the same CubeSat, and the third section addresses the expert insight of overall failure rates and causes for CubeSats, in general. This chapter focuses on the second section and uses part of the answers of the first section to provide a general context. Many questions had predefined multiple choice/selection answers as well as open boxes for alternatives and/or clarifications. Care is taken to avoid that the predefined options would bias the outcome and this has been verified by a small test panel which is not directly involved in this study. The questionnaire closed on the 1st of January 2015.

There were in total 138 participants of which 113 have fully completed the questionnaire. For 13 missions, there were multiple participants per CubeSat. In this case, the CubeSat-related answers are analysed and merged to a single answer. Wherever required, voting (for multiple choice questions with at least three participants), averaging (for numeric questions), or analysis (using publicly available information) have been applied to merge the answers. Following this process, there are answers on 60 launched CubeSats and 44 which are in the development or awaiting launch. In some cases, contradicting answers could not be merged into a single unbiased answer and those answers are excluded from further analysis. For some questions, the answers are unclear or skipped by the participant. In this chapter, the number of CubeSats which have been analysed for each question is provided with n.

The participation on 60 launched CubeSats is 24% of the total CubeSats deployed into orbit up to end 2014 and 29% if the 48 Flock CubeSats from Planet Labs deployed into orbit in 2014 are only counted as one satellite. In Figure 2.1, the participation distributed over time is provided and can be compared with the totals that are successfully deployed into orbit each year. The coefficient of determination for this aspect is calculated to be $R^2 = 0.76$ ($R^2 = 1$ would yield a perfect sample distribution). However, with the 2014 Flock satellites only counted once, $R^2 = 0.93$. Likewise, for the organizational type (educational, civil, military, and commercial), $R^2 = 0.59$.

2.2.2. General CubeSat Reliability

The participants have been requested to categorize their CubeSat mission in one or more objectives and provide success rates for each category. Of the 60 launched CubeSats, the provided mission objective categories are education ($n = 49$), technology demonstration ($n = 51$), science ($n = 27$), commercial ($n = 3$), civil ($n = 4$), military ($n = 1$), and radio amateur service ($n = 2$). In further investigation, sci-

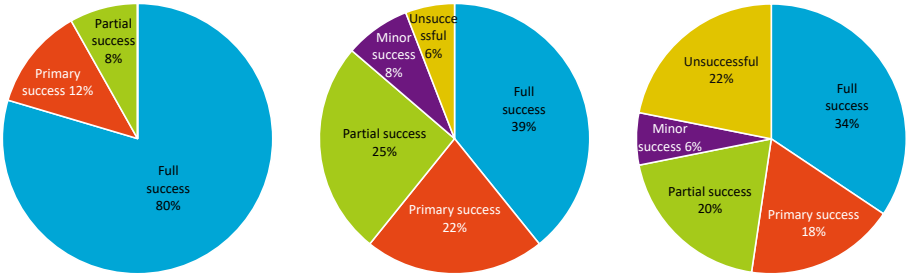


year:	'00	'01	'02	'03	'04	'05	'06	'07	'08	'09	'10	'11	'12	'13	'14
Survey Response				3			3		3		5	6	6	23	11
Total Deployed	5	1	1	6		3	6	7	6	11	15	9	23	78	83

Figure 2.1: Survey response and total CubeSats in orbit, distributed over launch years.

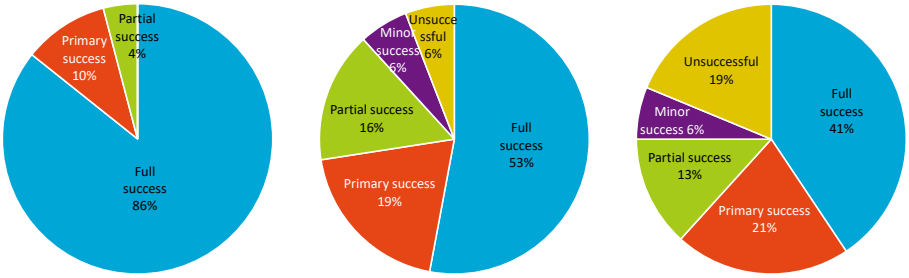
ence, commercial, civil, military, and radio amateur service objectives are grouped as 'operational' mission objectives ($n = 32$). The success rates are defined and ordered as unsuccessful, minor success, partial success, primary success, and full success. The results are provided in Figure 2.2 and 2.3. They are split in the current status and the expected final status. In the current status, missions which are not completed are also included and the overall success rates can, therefore, be lower than what potentially can be achieved. The expected final status is the most likely result (as foreseen by the participant) taken the current status into account and the potential future achievements for ongoing missions. Many missions have multiple objectives in different categories. The combination of education and technology demonstration is very popular ($n = 42$), while there are 20 missions with objectives in all three categories and only 10 missions in a single category.

For the educational success, many teams consider the successful launch of their satellite and/or minimal operations as a success of the educational objective. For technology demonstration, a minimum functionality of the satellite in orbit is required and it can be clearly seen that the success rates are lower for this objective compared with education. The demonstrated systems might, however, still not have to work completely according to specifications. For operational missions instead, the satellite really needs to work according to specifications to have a full success. For these most demanding objectives, the success rates are even further down and a vast majority of those CubeSat missions do not fulfill this objective completely. Only for 28% of all missions ($n = 60$), all objectives have been met so far. The expectation, as answered by the participants, is that 35% will eventually achieve full success.



(a) education ($n = 49$) (b) tech. demonstration ($n = 51$) (c) operational mission ($n = 32$)

Figure 2.2: Success rates as of Nov. 2014 of launched CubeSats for different main objectives.



(a) education ($n = 49$) (b) tech. demonstration ($n = 51$) (c) operational mission ($n = 32$)

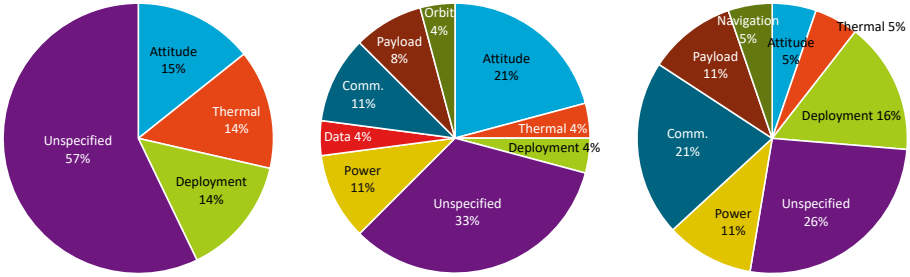
Figure 2.3: Expected final success rates of launched CubeSats for different main objectives.

The allocation of failures or issues to subsystems for not meeting one or more of the mission objectives with full success is investigated and its results are provided in Figure 2.4. The point of reference is the expected final success rates from Figure 2.3 and includes fully determined root causes of failures as well as observed anomalous symptoms of subsystems which could be used to provide a hypothesis for the subsystem which caused failure. For this study, it is only important to know which subsystem caused the satellite and/or mission failure. The following question was asked to the participants:

If applicable, have you been able to clearly identify a specific root cause for not fulfilling all mission goals or for the satellite failure? If yes, please specify:
O yes: (open)
O no, but we have one or more hypotheses: (open)
O no
O our satellite fulfilled all mission goals

The full set of provided answers can be found in the publicly disclosed database [10]. A few examples of this, with their subsystem allocation between brackets, are:

- *Yes: polarity of magnetic sensor became inverted. [attitude determination and control]*
- *Yes: degradation of solar cells, which finally lead to negative energy budget. [electrical power subsystem]*
- *No, but we have one or more hypotheses: on-board boot code was overwritten during a hardware reset or software crash or both [command and data handling]*
- *No [unspecified]*



(a) education ($n = 7$) (b) tech. demonstration ($n = 24$) (c) operational mission ($n = 19$)

Figure 2.4: Subsystem allocation for not achieving full success for different main objectives.

The electrical bus interfaces, which is the main topic of this study, are typically functionally allocated to the EPS and CDHS. Only for a single case, a hypothesis is provided that the data bus has been the root cause of not meeting one of the mission objectives. The EPS problems are allocated to batteries, solar cell degradation, and electronic malfunctions at the central EPS. There is no hard evidence that electrical interfaces have contributed to the mission failures of past and ongoing missions. It should, however, be noted that catastrophic failures on the critical electrical interfaces may lead to a sudden loss of satellite operation, which complicates root cause analysis. In addition, operational failures may have occurred after the mission design lifetime. Issues on the electrical interfaces which are non-catastrophic may still complicate operations and/or degrade the potential mission return. The potential correlation with catastrophic failures and other issues and electrical bus interfaces is investigated further in the following sections.

The mission design lifetime for the launched CubeSats ($n = 59$) is up to 6 months for 41%, between 8 and 12 months for 42% and more than a year for the remaining 17%. The average mission design lifetime is 11 months with a standard deviation of 9 months, a minimum of 1 month and a maximum of 5 years.

Figure 2.5 provides the operational status after a selected set of months since launch. The total percentage gradually drops with lifetime, as not all CubeSats have been in orbit for so long. Of the 14 CubeSats (23% of total) which have lost contact within 3 years, 12 CubeSats (20% of total) have not been operational for their entire mission design lifetime. This partially explains why so many technical

mission objectives have not been fully successful. The maximum reported fully operational status (including commanding capability) is 10 years and the average so far is 1.2 years.

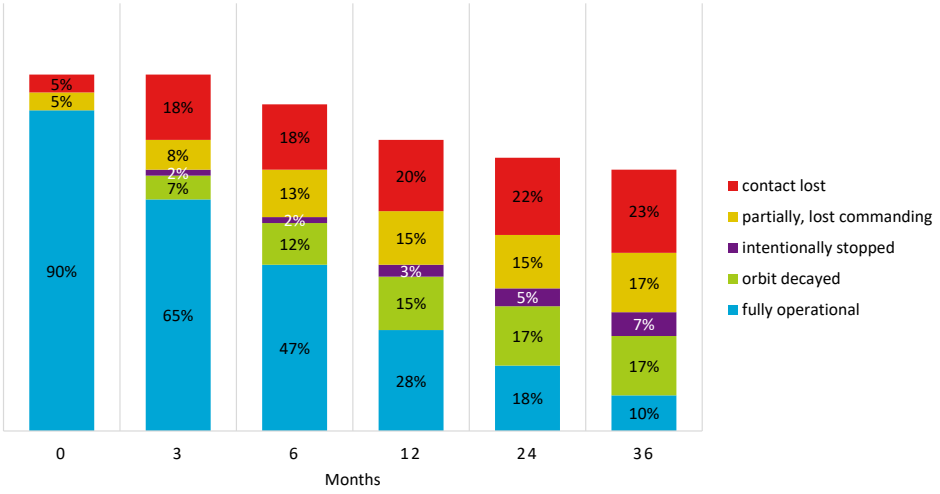


Figure 2.5: Operational status of CubeSats over time in-orbit, censored at Nov. 2014 ($n = 60$ at $t = 0$).

2.3. Survey Results on Data Busses

In Section 2.3.1, the data buses implemented are discussed. The data bus reliability is discussed in Section 2.3.2.

2.3.1. Data Bus Implementation

In Figure 2.6, the types of data interfaces that are implemented in the CubeSats between the main subsystems and/or main components are shown. The data buses are analysed only on the physical layer, which is layer 1 of the Open Systems Interconnection (OSI) model [11]. Local interfaces on a printed circuit board, for instance between a micro-controller and its peripherals, are not analysed. Multiple different data buses can be implemented on a single CubeSat: 45% have one, 22% two, 29% three, and 4% four ($n = 82$). The most frequently employed bus is the Inter Integrated Circuit (I^2C) bus. This is explained by the fact that many COTS integrated circuits have an internal I^2C controller and it is implemented in the majority of the COTS CubeSat subsystems. I^2C is a two-wire serial interface which connects two or more nodes in a master(s)-slave(s) configuration with typical data rates of 100 kbit/s for the standard mode and 400 kbit/s for a fast mode [12]. Practical experience has shown that a minimum clock frequency of 10 times the baud rate is needed for reliable operation within micro-controllers [6]. The maximum bus length depends on capacity, shielding, and additional buffering [12], but the protocol is designed for short distances and, in practical cases, is limited to several

tens of centimeters (many nodes using stacked connectors) up to several meters (few nodes, good wiring conditions).

Serial Peripheral Interface (SPI) is also widely implemented in CubeSats. SPI is a four-wire full duplex serial interface which connects one master with one slave at a time [13]. One line is a slave select, which can be multiplied from the master to connect multiple slaves, and is a form of physical addressing. The data rates are not bound by modes and can be several orders higher than for I^2C. However, the slave needs to handle the throughput of the master without the ability to stretch the clock as with I^2C, otherwise the data can be lost. Similar to I^2C, this bus is designed for short distances up to several meters in the favourable conditions.

Recommended Standard 232 (RS-232) scores about equal to SPI in implementation rate. RS-232 is a serial data interface between digital systems and is full duplex [14]. It cannot be connected or expanded to more than two nodes. Its raw data rates (including protocol overhead) are up to 20 kbit/s according to the standard. However, in current practice, non-standard 3-wire and 5-wire implementations up to 115.2 kbit/s are used. The standard is designed for distances up to 15 m [14].

The controller area network (CAN) bus has been implemented only in a few launched CubeSats, but is becoming more popular. CAN bus is developed for automotive applications and is designed to be able to operate in harsh environments. It is a serial bus with differential signalling and failure tolerance at OSI layer 1 is possible for this bus. Data rates are up to 1 Mbit/s. Cable connections can span a distance up to 40 m.

Universal serial bus (USB) shows a similar trend as the CAN bus. USB is the currently most popular standard to connect computers with peripheral equipment. It has several backwards compatible modes with the current fastest being SuperSpeed + (USB 3.1) which supports 10 Gbit/s of data rate [15] over 8 wires (full duplex). USB can only be used to connect one master to one slave, which can be extended using a hub.

Universal Asynchronous Receiver/Transmitter (UART) is no exclusive data bus, but a piece of (integrated) hardware to manage the link between a microprocessor and a serial data bus and includes RS-232, RS-422 and RS-485.

Space-Wire is the only data bus designed specifically for space applications. It is, however, only implemented in one CubeSat which is still to be launched. In addition, wireless standards are not widely implemented yet.

2.3.2. Data Bus Reliability

The questionnaire addressed the reliability of the implemented data buses. To exclude immature designs and to have sufficient statistical input, only I^2C, SPI, and RS-232 for launched CubeSats are analysed in this section. The results for the implemented failure tolerance features at OSI layer 1 [11] for each bus is presented in Figure 2.7. Except for the optional error line for I^2C, the three buses do not have inherent failure tolerance at OSI layer 1. This means that most failure tolerance features presented in Figure 2.7 are designed and implemented by the CubeSat developers. Single-wire failure tolerance means that the bus is still operational if there is a fault on one line. This can lead to maintained performance (e.g. in case of

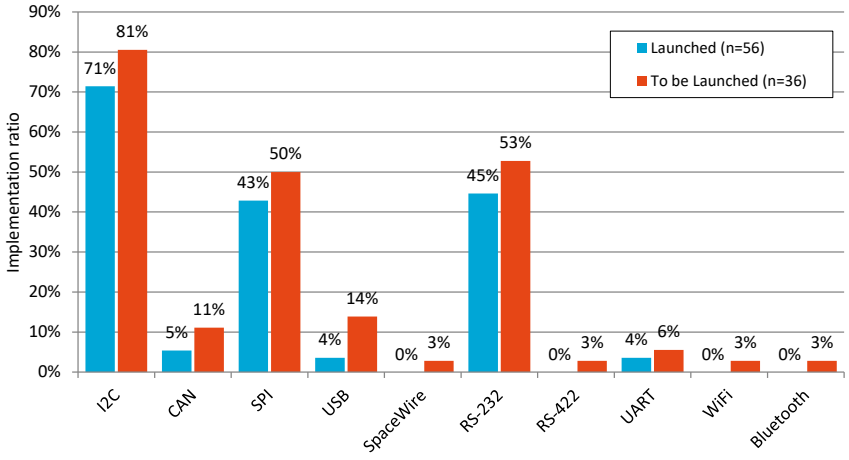


Figure 2.6: Implemented data buses in CubeSats.

full redundancy) or to degraded performance (e.g. going from differential signalling to single-ended signalling by lowering the maximum data rate).

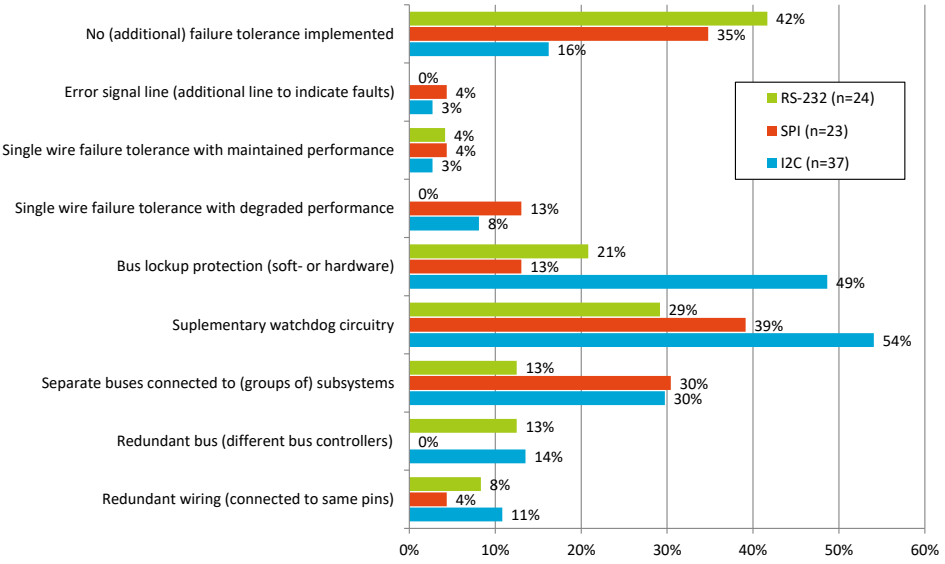


Figure 2.7: Implemented data buses in CubeSats.

It can be clearly seen from Figure 2.7 that the majority of CubeSats have additional failure tolerance features implemented for the three different buses. Very popular are bus lockup protection and supplementary watchdog circuitry (these fea-

tures may be overlapping). For SPI, there are also many failure tolerance features implemented, although significantly less than for I^2C. For both buses, however, in about 30% of the cases, there is not just a single serial bus connector to all subsystems. Instead, there are separated buses connected to (groups of) subsystems of the same bus type. Redundant wiring and buses are not implemented widely, potentially due to limited availability of micro-controllers and peripheral devices with dual I^2C or SPI controller and the complexity of such solutions. The RS-232 interface has the least amount of failure tolerance features implemented, but this bus standard can only connect two nodes. For 87% ($n = 24$) of the CubeSats, this is complemented with one or more of the other bus types. Figure 2.8 presents the reported in-orbit issues for I^2C, SPI, and RS-232. For I^2C, bus lockups appear to be a major issue. Also for one CubeSat, I^2C has led to a catastrophic failure (proven), while for two more I^2C are a likely cause (hypothesis). For RS-232 and SPI, only few issues are reported. There are several possible explanations why I^2C has a relatively high amount of reported issues:

- I^2C lacks separate lines for handshaking and control compared with SPI and RS-232.
- For I^2C, typically a higher number of nodes is connected to the same bus than for SPI and RS-232.
- I^2C does not have differential signalling and is implemented without shielding in most CubeSats.
- The state-machine of I^2C hardware controller and firmware may have errors [6]. This risk becomes larger when different micro-controllers are connected to the same bus because they have have slightly different implementations, for example on time-out handling.

Some examples of I^2C problems are found in literature. The failure of the CP4 CubeSat of California Polytechnic State University is most likely the result of I^2C data bus problems [16]. The Delfi-C^3 satellite of TU Delft also experienced major I^2C problems, which caused high bit-error-rates as well as sustained bus lockups [6]. These bus lockups were recovered by the power reset in each eclipse, because of the absence of a battery in this satellite. The Delfi-n3Xt satellite of TU Delft lost its transmission signal[1] after completion of its primary mission objectives when an experimental transponder was switched on [17]. The main hypothesis for the root cause is that an I^2C buffer has been blown up by the transponder, effectively terminating all internal data handling. During a late development stage, it was already discovered that the implemented I^2C buffer has a potential failure mode which shorts the bus to ground.

[1] In February 2021, after seven years of silence, Delfi-n3Xt has spontaneously started transmission again. Pending further investigation, the main hypothesis is still considered plausible.

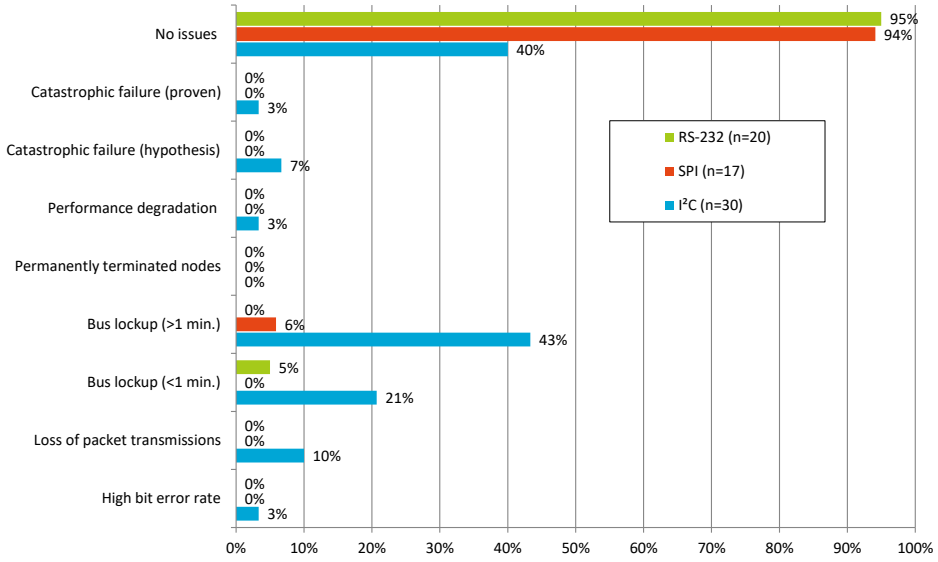


Figure 2.8: In-orbit issues reported for three bus standards.

2.4. Survey on Electrical Power Interfaces

2.4.1. Electrical Power Subsystem Implementation

A central electrical power subsystem (EPS) unit is typically responsible for the power conversion from the solar panels, the energy storage, as well as the distribution of power to the rest of the satellite over one or more power bus lines. The ability to switch subsystems on and off, over-current protection, and the power bus topology have an impact on the power bus reliability. Of the combined launched and to be launched CubeSats ($n = 84$), 63% have a custom designed central EPS unit, 19% a COTS unit from GomSpace, 15% from ClydeSpace, 1% from CubeSat Kit, and 1% from Tyvak. In Figure 2.9, the topology and reliability features implemented on the CubeSats are provided. A shared line means that multiple subsystems are drawing current from a single power line. If this line is only protected at the central EPS unit, the subsystems allocated to this shared line pose a threat to each other. If they are, however, protected at the subsystems locally, this risk is mitigated. For individual lines to each subsystem, protection could be at the central unit and/or the local subsystem. As example for such power protection, in the MicroMAS CubeSat of MIT, Fairchild FPF2700 high-side current limit switches are implemented to provide load switching and over-current protection [18]. Another example is the protection circuit within Delfi-n3Xt which uses a dedicated circuit at each subsystem locally to protect both the main power bus as well as the I^2C data bus [19].

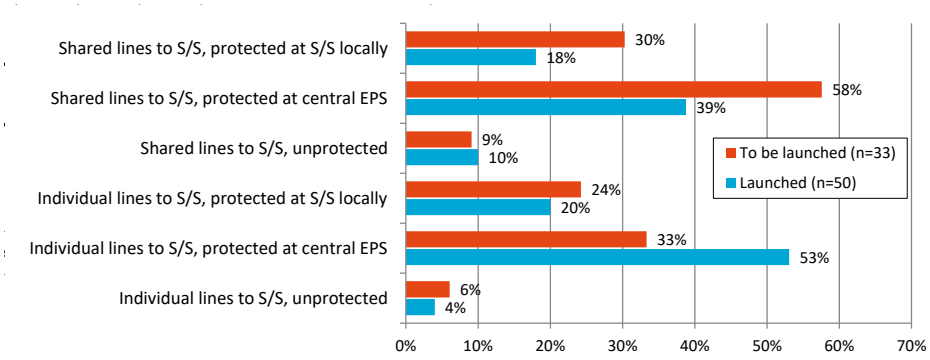


Figure 2.9: Electrical power distribution topology and reliability features.

2.4.2. Power Distribution Reliability

A 29% of the launched CubeSats ($n = 49$) have reported in-orbit issues with the EPS. The issues are categorized and provided in Figure 2.10.The rapid solar cell degradation (n = 3) is noteworthy, as two reported this was due to missing cover glasses. For fully unprotected distribution lines, there is a risk to the overall EPS. A 10% of the CubeSats ($n = 50$) have implemented unprotected shared distribution lines. One of them lost contact after 10 days, with a hypothesis of a single event effect caused by ionizing radiation resulting in a short circuit. A second CubeSat out of this five never made contact, for which the reason is unknown. Despite small sample size, these two examples exhibit the risk of implementing unprotected power distribution lines.

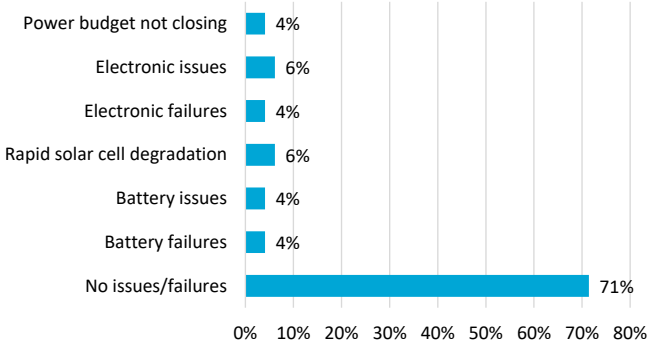


Figure 2.10: In-orbit issues reported for the EPS ($n = 49$).

Several papers address the relation of power distribution protection features and radiation effects. It was reported that Cute-1.7 + APD have lost operability of their satellite about 3 weeks after launch, most likely due to radiation effects [20]. On ground, analysis revealed that a combination of a latch-up and a too high threshold for an over-current protection circuit have caused damage to the on-board

integrated circuits [20]. The GOMX-1 satellite reported (intentional) power cycles of their payload triggered by bit-flips in the field-programmable gate array (FPGA) due to radiation effects, many of them occurring in high-energy proton regions, such as the South Atlantic Anomaly (SAA) [21]. However, for GeneSat-1 analysis of in-orbit data of the performance of Peripheral Interface Controllers (PICs) indicated no CPU resets, latch-ups or single-event upsets during its 18 months of operation [22].This can be explained by the different orbit of GeneSat-1, a circular orbit with an altitude of 460 km and an inclination of 40.5º, while GOMX-1 had a Sun-Synchronous Orbit (SSO) with an altitude between 600 km and 835 km and an inclination of 97.8º. Satellites in SSO are more challenging in terms of radiation environment due to the passages over the polar regions where the Earth magnetic field is weaker and therefore the flux of galactic cosmic rays and solar particles is higher. Another possible reason for the different radiation hardness performances of the 2 satellites is related to the fact that the susceptibility to radiation effects is strongly dependent on the specific type of integrated circuits implemented.

2.5. Electrical Power Subsystem Architecture

To be able to understand the distribution of electrical power, a few commercial electrical power subsystems and a custom one are investigated using the Delfi-n3Xt satellite as reference case.

2.5.1. Electrical Power Subsystem Architecture

The power distribution of two commercial EPS and one custom EPS has been studied: the GomSpace P31u [24], the ClydeSpace CS-3UEPS2-NB [25] and Delfi-n3Xt EPS [19]. Their power distribution is provided schematically in Figures 2.11, 2.12 and 2.13.

Both commercial systems have in common that there are three outputs available to the subsystems: 3.3 V, 5 V and a variable battery bus. The difference is that the GomSpace P31us version supports a higher variable battery bus voltage range for connecting four Li-ion batteries in series. Also switchable and current-monitored outputs can be configured on the GomSpace EPS, which can connect individual subsystems directly with their own supply line. This however may lead to incompatibilities with other commercial CubeSat systems as the PC/104 pin assignment for these switched lines is not standardized.

On Delfi-n3Xt, an Electrical Power Subsystem (EPS) is developed in cooperation between TU Delft and SystematIC design BV. The power from the solar panels is converted with Maximum Power Point Trackers (MPPTs) to a variable voltage bus of 19-30 V. This variable bus is connected to a 12 V regulator board and the battery system. The Delfi-n3Xt is equipped with Li-ion batteries for energy storage. When the variable voltage level is dropping below 19 V, the batteries supply this bus to maintain this voltage level by up-conversion from the Li-Ion battery voltage of 3.0 V to 4.2 V. When the level is above 22 V, the batteries are being charged at a rate related to the voltage level. Above 26 V, the MPPTs gradually step away from the

Parts of Section 2.5 have been published in the proceedings of the 4S conference 2014 [23].

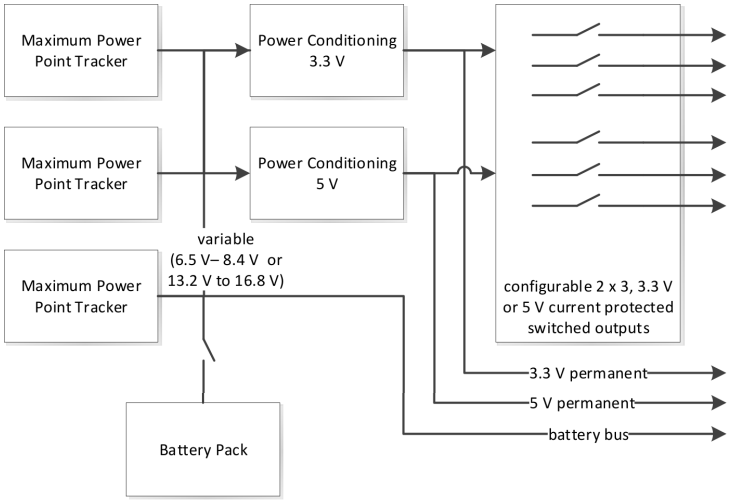


Figure 2.11: GomSpace P31u schematic.

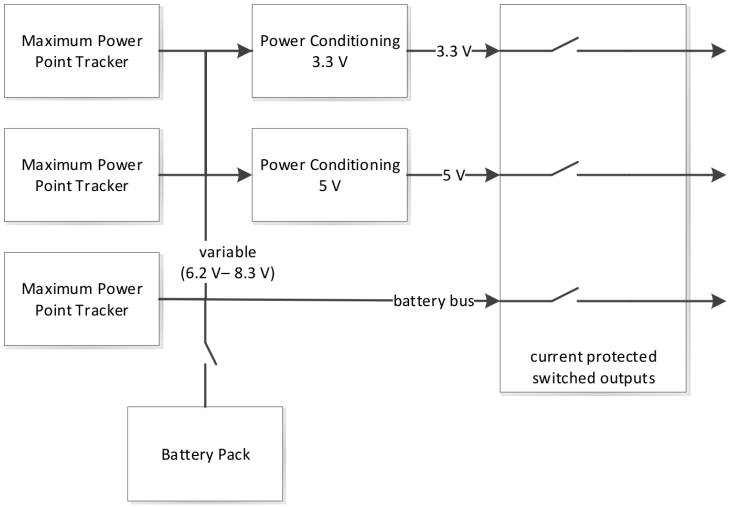


Figure 2.12: ClydeSpace CS-3UEPS2-NB schematic.

maximum power point. For transient peak power available, a safety shunt can temporally clamp the variable voltage at 30V to avoid this bus being up converted to extreme levels where components may be damaged. All other subsystems use the 12 V single power supply line. The heritage of Delfi-C^3 [26] was one of the main reasons to choose this voltage level. Most of Delfi-n3Xt subsystems require 3.3 V which is down-converted locally at the subsystem with DC-DC converters. Some high power consuming devices such as the deployment circuit however directly uses the 12 V bus. Investigation shows however that this voltage level can easily

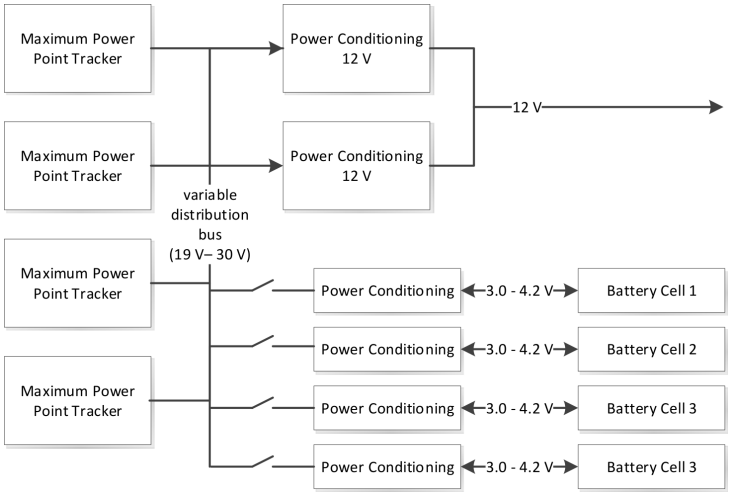


Figure 2.13: Delfi-n3Xt EPS schematic.

be changed to another fixed or variable level beyond 6 V without major redesign if desired.

In Delfi-n3Xt, battery charging and discharging is performed by down-conversion from and up-conversion to the variable distribution bus. The four Li-ion batteries are of 18650 type with individual voltages of between 3.0 V and 4.2 V. This type of cells are also used in the commercial EPS systems [24, 25]. For a long cycle life time of several years, the nominal orbital depth-of-discharge should be limited to about 10-20%, which means that the voltage would be between 3.8 V and 4.2 V typically. In Delfi-n3Xt, the Li-ion batteries were charged according to a Constant Current / Constant Voltage scheme to protect the batteries from over-current and over-voltage respectively. In practice however, the excess power of a CubeSat is limited while the individual batteries can support up to 2 A of current, which would yield 4 W to 6 W of power per battery at 3 V and 4 V respectively. Over-current protection would be possible by simply having sufficient batteries connected either in series or parallel. This would provide the possibility to have a variable battery bus which directly connects to both batteries as well as any other subsystem using current from that line, as is shown in Figures 2.11 and 2.12 of the commercial EPS solutions. If there is excess power available, the MPPTs can increase the variable bus voltage above the battery voltage until a balance between available power and power consumption (inc. battery charging) is reached. Likewise, if there is insufficient power from the solar panels, the variable voltage drops below the battery voltage and the batteries will supply current to the bus. This yields the same advantage of having an agile distribution bus as in the Delfi-n3Xt EPS architecture, but without the additional conversion losses. Furthermore it is recommended to have the capability to attach/detach each series of batteries from the bus for failure tolerance and potential implementation of a kill switch to keep the satellite unpowered during launch.

Operating the battery bus voltage at individual level would yield relatively high currents and losses in case of peak power excess or consumption. It is therefore logical to connect two batteries in series to have a typical range of 7.4 V to 8.4 V. Connecting four of them in series means increase battery system risks and might be over-dimensioned for small satellites. Other subsystems can make use of the battery voltage level as well and this is recommended for high power consuming elements such as power amplifiers and heaters

2.5.2. Electrical Power Interfaces to Subsystems

On Delfi-n3Xt all subsystems are supplied by a 12 V single supply voltage and converted to lower voltages locally. A small survey is performed on a random selection of commercially available CubeSat systems and presented in Table 2.1. The information is obtained from the company brochures in April 2014.

Table 2.1: Power supply interfaces of selection of commercial CubeSat systems.

Brand	Model	Subsystem	Interface	Supply line(s)
Argus	100	Infrared P/L	Custom	3.6-4.2 V
ClydeSpace	CMCi UVTRX	Radio Trx.	PC/104	6-9 V
ClydeSpace	STX	Radio Trx.	PC/104	7.2-12 V
GOMspace	NanoCom U482	Radio Trx.	PC/104	3.3 V
GOMspace	NanoMind A712D	OBC	PC/104	3.3 V
GOMspace	NanoCam C1U	Camera P/L	Custom	3.3 V
IQ-Wireless	Hispico	Radio Tx.	Custom	3.3 V
ISIS	TRXUV	Radio Trx.	PC/104	6.5-12.5 V
ISIS	TRXVU	Radio Trx.	PC/104	5-18 V
ISIS	TXS	Radio Tx.	PC/104	3.3, 5 & 6.5-30 V
Maryland	MAI-100	ACS	Custom	12 V
MicroSpace	Micropropulsion	Propulsion	Custom	12 V
Pumpkin	MB	OBC	PC/104	7-10 & 5V

From this survey some additional observations are made:

- Most CubeSat systems adopt the PC/104 board-to-board interface standard.
- Some radio transceivers have a relation between the used (raw) supply voltage, the performance and the power consumption.
- Cubesat systems use a heterogeneity of input supply voltages, but typical inputs are 3.3 V, 5 V, 12 V and variable battery busses which has 7 V to 9 V as general overlap.
- Most Cubesat systems use a single supply voltage and convert to other voltages locally if needed.
- Many commercial systems offer different voltage levels and pin assignment in case of a PC/104 interface. Some pins of the PC/104 are used for different purposes which leads to incompatibility problems when combined.
- For the PC/104 standard, typically three lines are allocated as power ground.

2.5.3. Power Losses

For the PC/104 standard, the connectors used are board-to-board stackable pin headers of four rows of 26 pins each, which are pitched at 0.1" or 2.54 mm. To consider the wiring loss, the connector resistance and the pin resistance need to be taken into account. Considering a copper resistivity of $\rho = 1.7 \cdot 10^{-8} \Omega m$ at 20°C and taking the 0.64 mm x 0.64 mm dimensions of the pins into account, the resistance is $R(l) = 0.041 \Omega/m$. The pin resistance of three randomly picked gold plated header specifications of different manufacturers, namely the Samtec SSQ series, 3M 4P series and Harwin M20-610 series provide 10 mΩ typical, 20 mΩ maximal and 30 mΩ maximal respectively.

A measurement was performed with the Samtec connectors. In the measurement seven connectors were stacked with a total length of 7 cm and a current of 0.50 A and 1.00 A was applied with a current limited power source. A voltage over the connectors of 17 mV and 32 mV was measured respectively. This means a total resistance of 33 mΩ and a contact resistance per connector of 4.4 mΩ (pin resistance subtracted). This is significantly lower than specified. The difference between the measurement and also between the different specifications might be caused by the amount of duty cycles of the connector applied, in which contact resistance will gradually increase. It is therefore recommended to limit the amount of cycles, e.g. by using pin savers during subsystem testing.

In Delfi-n3Xt there were 11 PCBs connecting to the main interface bus. In Table 2.2 a case study of Delfi-n3Xt is investigated if this satellite would have used the PC/104 standard. The distance between boards D and the amount of connectors in between is taken into account. For the resistance R, the return path to ground is included and it is assumed that this is distributed over three lines as also observed for commercial subsystems.

Table 2.2: Wiring resistance and power loss of Delfi-n3Xt in case of PC/104 stacked connectors.

Subsystem				**Loss when operated at:**		
Unit	**D [mm]**	**R [mΩ]**	**P [mW]**	**3.3 V**	**5 V**	**12 V**
Payload	17	13	83	0.01%	0.00%	0.00%
Attitude control	21	13	1649	0.19%	0.08%	0.01%
Propulsion	62	27	45	0.01%	0.00%	0.00%
Computer	106	41	232	0.09%	0.04%	0.01%
Primary radio	127	54	1745	0.86%	0.38%	0.07%
S-band radio	155	67	121	0.07%	0.03%	0.01%
Secondary radio	176	80	254	0.19%	0.08%	0.01%
Battery control	233	95	2538	2.21%	0.96%	0.17%
Temp. sensors	246	107	82	0.08%	0.04%	0.01%
Full satellite			6749	1.11%	0.48%	0.08%

From Table 2.2 can be concluded that the wiring loss in all cases is acceptable for nominal operational modes. On Delfi-n3Xt, flex-rigid wiring is implemented in combination with side-mounted Harwin Datamate connectors. Similar to the

PC/104 connector, the contact resistance is measured and is found to be 6 mΩ for a connector which is not mechanically cycled many times. The wiring are tracks on a flex-rigid PCB and the connectors are soldered to this tracks. The advantage here is that there are always only two connectors in between the EPS board and the subsystem. The copper layers of the flex-rigid PCB are however limited in dimensions. Each track is 0.4 mm wide and has a thickness of 35 µm, yielding $R(l) = 1.21\Omega/m$ per track. As there are four redundant 12 V tracks applied, the resistance becomes $R(l) = 0.30\Omega/m$ for this bus. The ground is a fully meshed plane with an effective width of 12mm, yielding $R(l) = 0.04\Omega/m$. Applying the same method as presented on Table 2.2 for this wiring harness, yields a total loss of 1.31% for 3.3 V, 0.57% for 5 V, and 0.10% for 12 V. There is thus no major difference between stacked connectors or flex-rigid wiring in terms of power losses. The main advantages of flex-rigid wiring is that the connector is mounted on the side and potentially saves some board space over a stack-through connector and can mechanically decoupled to increase thermo-mechanical cycle life-time.

2.6. Survey on CubeSat Wiring Harness

The connector of the PC/104 bus standard has become the de-facto standard wiring harness in CubeSats, as most commercial developers provide their subsystems with this interface. The PC/104 bus standard is derived from the ISA bus applied in personal computers in the past, and its initial release was 1992 [5]. It defines not only the wiring harness, but also the form factor of the printed circuit boards and their mechanical mounting [5] and is supposed to be able to stack embedded systems on top of each other. In Figure 2.14, an example is provided showing the 104-pin stack-able connector.

Figure 2.14: Example of a CubeSat on-board computer (TU Delft) with PC/104 connector.

For CubeSats, many aspects of the PC/104 standard are not implemented. The commercial suppliers of CubeSat subsystem typically implement the I^2C data bus, while the Industry Standard Architecture (ISA) bus is the intended data bus for the PC/104 standard [5]. In addition, the allocation and distribution topology for power are not taken over, nor standardized for CubeSats, leading to compatibility issues as described in Section 2.5.1. Therefore, when PC/104 is mentioned as standard in relation to CubeSats, this refers to a fixed physical wiring harness and the mechanical layout and not the data bus or pin allocation.

For all CubeSats, the specification or standard of wiring harness was asked in the survey. These are grouped in 'PC/104' and 'other', as no other standard is clearly implemented by multiple developers. For the launched CubeSats, 35% have implemented the PC/104 connector ($n = 49$). For the CubeSats not yet launched, this is 59% ($n = 32$), probably due to the increasing availability of commercial subsystems with a PC/104 connector. For CubeSats with different wiring harness, the information provided is too limited and heterogeneous for statistical analysis.

Of the launched CubeSats, there are no in-orbit failures reported which are directly allocated to the wiring harness such as wiring breaks or short circuits. The standard itself, however, partially drives power bus distribution and data bus selection, which is already treated in the previous sections. In Figure 2.15, the design issues for all CubeSats (launched and to be launched) with the PC/104 connector are provided based on simple 'yes or no' questions. The wires are extended through-hole pins which can be stacked into the next connector. In some cases, an additional connector is stacked in between two printed circuit boards to be able to span a wider distance.

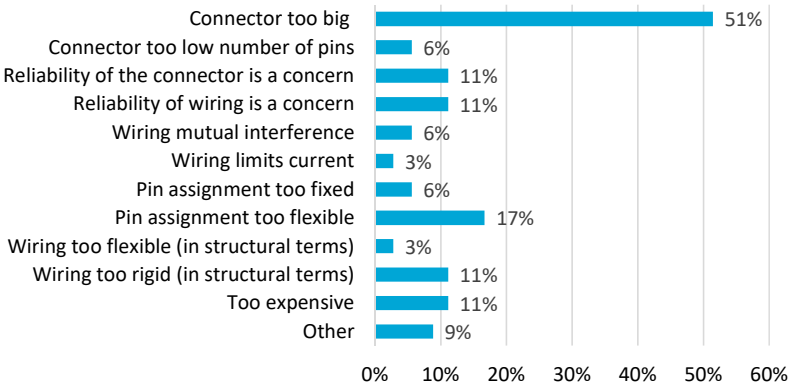


Figure 2.15: Design issues reported on PC/104 connector ($n = 36$).

It can concluded from Figure 2.15 that a (small) majority stated that the PC/104 connector is too big, while only a small minority considers the number of pins to be too little. Thus, for a future standard, a smaller connector with fewer pins is recommended. The reliability of the connectors or wiring is a concern for a minority of the participants. Three additional issues ('other' in Figure 2.15) are reported: I^2C

cannot handle long wiring, there are difficulties with mating/de-mating during the development phase and PC/104 structural layout does not create good thermal paths.

From literature, a few additional observations are made. According to Kimm and Jarell [27], the PC/104 CubeSat Kit from Pumpkin shows two weaknesses: the non-redundant architecture of the on-board processor and the fairly large size of the physical bus interface. It is also noted that the form factor makes it difficult to include redundant systems. They suggest to use CAN for a data bus in CubeSats and propose a protocol (at higher OSI layers) to handle data traffic. During the development of the EDSN satellite, it was found that low-cost consumer grade connectors sometimes provided physical alignment issues and even intermittent contacts. Noticeably these issue can be avoided by using higher grade components, a practise which is strongly recommended [28].

2.7. Conclusions on Survey

In this chapter, the implementation and reliability of electrical bus interfaces are analysed through a literature survey and a questionnaire. In this section, the main results and conclusions are provided.

It has been found that at least 65% of the CubeSats in this survey ($n = 60$) are expected not to fulfill all mission objectives. There is, however, no evidence that the electrical interfaces are a major cause for this. The fully operational lifetime of CubeSats is 1.2 years on average, while 20% of the CubeSats have not been operational for their design life time.

Currently, the most popular buses for CubeSats are I^2C, SPI, and RS-232. The I^2C data bus shows many bus lockup issues on CubeSats (for three CubeSats, this bus is likely to have caused catastrophic damage, of which two beyond the mission design lifetime. Especially, the combination of unshielded pins of the PC/104 connector and the I^2C data protocol is a concern.

While the EPS is a major source of in-orbit failures, most of those failures are not allocated to power bus interface. However, of the five CubeSats that have implemented no protection of the power distribution lines, two have failed after some days in orbit.

In terms of electrical power subsystem architecture, it is observed that different suppliers use different power distribution in terms of supply lines and voltages. Power losses through stacked connectors, or alternative wiring, are limited and should not be driving factor in trade-offs. The number of voltage level conversion steps in series can, however, best be limited for maximum efficiency.

For the majority of the CubeSats which implemented the PC/104 connector, it is stated that the connector is too large, but no catastrophic in-orbit failures have been reported, because of this connector.

For a future CubeSat bus standard, the following recommendations can be made:

- The data bus should provide continuity: disruptions causing temporarily unavailability (e.g. bus lockups) should occur less than ten times per day in

nominal conditions and should always recover in less than 10 seconds.

- The power distribution lines should protect both the central EPS unit as well as the local subsystems against short circuits and over-current, including those induced by radiation effects.
- The standard interface connector should be smaller than the PC/104 connector.
- The standard interface should have a fixed pin allocation to achieve general compatibility between subsystems.
- The standard interface should allow limited voltage conversion steps in series in the overall EPS architecture.

An important aspect for a future bus standard is the overall redundancy and failure mitigation concept for CubeSats. If there will be redundant physical subsystems, the bus standard may need to accommodate failure detection, isolation, and recovery with specific wiring and/or circuitry. In addition, the data and power interfaces themselves may require redundancy. Further study into the needs and analysis on the impact of this aspect is recommended for the development of a new CubeSat bus interface standard. For improved analysis on this aspect and for CubeSats, in general, it is recommended that CubeSat developers publish more details about their in-orbit results, with an emphasis on the experienced issues, anomalies, and failures.

References

[1] J. Bouwmeester, M. Langer, and E. Gill, *Survey on the Implementation and Reliability of CubeSat Electrical Bus Interfaces,* CEAS Space Journal **9** (2017), 10.1007/s12567-016-0138-0.

[2] J. Bouwmeester and J. Guo, *Survey of Worldwide Pico- and Nanosatellite Missions, Distributions and Subsystem Technology,* Acta Astronautica **67**, 854 (2010).

[3] M. Swartwout, *The First One Hundred CubeSats : A Statistical Look,* Journal of Small Satellites **2**, 213 (2013).

[4] R. Munakata, *Cubesat Design Specification (rev.12)*, Tech. Rep. August 2009 (California Polytechnic State University, 2009).

[5] *PC/104 Specification (v2.5)*, Tech. Rep. (PC/104 Embedded Consortium, 2003).

[6] N. Cornejo, J. Bouwmeester, and G. Gaydadjiev, *Implementation of a Reliable Data Bus for the Delfi Nanosatellite Programme,* in *Proceedings of the 7th IAA Symposium on Small Satellites for Earth Observation* (IAA, Berlin, 2009).

[7] M. Tafazoli, *A Study of On-orbit Spacecraft Failures,* Acta Astronautica **64**, 195 (2009).

[8] S. Y. Kim, J. F. Castet, and J. H. Saleh, *Spacecraft Electrical Power Subsystem: Failure Behavior, Reliability, and Multi-state Failure Analyses,* Reliability Engineering and System Safety **98**, 55 (2012).

[9] J. Guo, L. Monas, and E. Gill, *Statistical Analysis and Modelling of Small Satellite Reliability,* Acta Astronautica **98**, 97 (2014).

[10] J. Bouwmeester and M. Langer, *Database: Results of CubeSat Survey on Electrical Interfaces & Reliability,* (2016).

[11] ITU, *ITU-T Recommendations X.200*, Tech. Rep. (ITU, 1994).

[12] NXP Semiconductors, *PCA9615 (Datasheet)*, Tech. Rep. 1.1 (NXP, 2016).

[13] *SPI Block Guide v3.06*, Tech. Rep. (Motorola, Inc., 2003).

[14] S. S. V. Rani, S; Malhotra, *Detailed Study of RS-232 Serial Interface,* International Journal of Research **1**, 1187 (2014).

[15] *Universal Serial Bus 3.1 Specification (v1.0)*, Tech. Rep. (Hewlett-Packard, Intel Corporation, Microsoft, Renesas, ST-Ericsson, Texas Instruments, 2013).

[16] G. Manyak and J. M. Bellardo, *PolySat's Next Generation Avionics Design,* in *Proceedings of the 4th IEEE International Conference on Space Mission Challenges for Information Technology* (IEEE, Palo Alto, 2011).

[17] J. Bouwmeester, L. Rotthier, C. Schuurbiers, W. T. Wieling, G. V. D. Horn, F. Stelwagen, E. Timmer, and M. Tijssen, *Preliminary Results of the Delfi-n3Xt Mission,* in *Proceedings of the 4S Symposium* (ESA, Mallorca, 2014).

[18] R. Kingsbury, F. Schmidt, W. Blackwell, I. Osarentin, R. Legge, K. Cahoy, and D. Sklair, *TID Tolerance of Popular CubeSat Components,* in *IEEE Radiation Effects Data Workshop* (IEEE, San Francisco, 2013).

[19] J. Bouwmeester, C. A. H. Schuurbiers, F. Stelwagen, M. Tijssen, and E. Timmer, *Delfi-n3Xt : A Powerful Triple-unit CubeSat for Education and Technology Demonstration,* in *Proceedings of the 2nd IAA Conference on University Satellite Mission and CubeSat Workshop* (IAA, Rome, 2013).

[20] K. Omagari, K. Konoue, N. Miyashita, M. Iai, H. Yabe, K. Imai, K. Miyamoto, S. Masumoto, T. Iljic, K. Fujiwara, T. Usuda, Y. Konda, S. Sugita, T. Yamanaka, and S. Matunaga, *Tokyo Tech Second Nano-Satellite Cute-1.7 + APD and its Flight Operation Results,* in *Proceedings of the 57th International Astronautical Congress* (IAF, Valencia, 2006).

[21] L. Alminde, K. Kaas, M. Bisgaard, J. Christiansen, and D. Gerhardt, *GOMX-1 Flight Experience and Air Traffic Monitoring Results,* in *Proceedings of the 28th AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2014).

[22] G. Minelli, C. Kitt, K. Ronzano, C. Beasly, R. Rasy, I. Mas, and P. Williams, *Extended Life Flight Results from the GeneSat-1 Biological Microsatellite Mission,* in *Proceedings of the 22nd AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2008).

[23] J. Bouwmeester and N. Santos, *Analysis of the Distribution of the Electrical Power in CubeSats,* in *Proceedings of the 4S symposium* (ESA, Valetta, 2014).

[24] Gomspace, *NanoPower P-series Datasheet v. 8.0*, Tech. Rep. (Denmark, 2013).

[25] A. Strain, *CubeSat 3U Electronic Power System CS-UEPS2-NB (issue A)*, Tech. Rep. (United Kingdom, 2011).

[26] R. Hamann, J. Bouwmeester, and G. Brouwer, *Delfi-C3 Preliminary Mission Results,* in *Proceedings of the 23rd AIAA/USU Small Satellite Conference* (AIAA, Logan, 2009).

[27] H. Kimm and M. Jarrell, *Controller Area Network for Fault Tolerant Small Satellite System Design,* in *Proceedings of the 23rd IEEE International Symposium on Industrial Electronics* (IEEE, Istanbul, 2014).

[28] J. Chartres, H. Sanchez, and J. Hanson, *EDSN Development Lessons Learned,* in *Proceedings of the 28th AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2014).

3

Lean Electrical Interface Standard

Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.

Antoine de Saint-Exupery

Developers of CubeSats and PocketQubes experience issues with the compatibility, connector size and robustness of electrical interface standards for their satellites. There is a need for a lean and robust electrical interface standard for these classes of satellites. The proposed interface standard comprises a linear data bus which is used for housekeeping data, internal commands and small-to-moderate payload data. A community based analytic hierarchy process is used for the trade-off of design options, resulting in the selection of RS-485 as standard data bus, mainly due to its low power consumption and high effective data throughput compared to other candidates. Several switched and protected battery voltage lines are distributed from the central electrical power subsystem unit to the other subsystems to enable a simple and efficient power distribution. The harness comprises a 14 and a 9 pin stack-able connector for CubeSats and PocketQubes, respectively, occupying very little board space.

The content of this chapter has been published in Advances in Space Research **62**, 12 (2018) [1]

3.1. Introduction

CubeSat and PocketQube developers experience issues with the compatibility, connector size and robustness of electrical interface standards. This chapter describes the process towards a lean electrical interface for CubeSats and PocketQubes which should tackle these issues. Subsequently, the results of an extensive trade-off for the electrical interfaces for PocketQubes and CubeSats are presented. The standard electrical interfaces typically comprise one or more digital data busses used for the transmission of data between subsystems and power distribution lines. Optionally, an electrical interface standard can also comprise lines for base-band radio signals, analogue signals and general input/output.

Based on design targets specified in Section 2.7, an appropriate standard data bus architecture is presented in Section 3.2. Section 3.3 describes the trade-off process and Section 3.4 provides the trade-off results for the data bus. Section 3.5 provides a brief analysis on power distribution. In Section 3.6 a new electrical bus interface standard for PocketQubes and CubeSats is proposed, which is lean and which facilitates efficient power distribution and ensures inter-subsystem compatibility. Finally, conclusions and a future outlook is provided in Section 3.8.

3.1.1. Motivation for a new electrical interface standard

In the survey on CubeSat electrical interfaces in Chapter 2, it became clear that many CubeSat developers experience issues with the de-facto standard electrical interface based on the PC/104 connector, part of the PC/104 standard and the I^2C data bus. Documents which describe the pin allocation for PC/104 connectors for CubeSats do not exist and it was previously found that subsystems from different commercial suppliers use different pin allocations.

A proposal for a dedicated CubeSat electrical interface standard comes from UNISEC [2]. It defines, amongst others, a standard 50-pin stacked connector between subsystems comprising power distribution at various voltage levels, several options for data interfaces (I^2C, UART, JTAG), reset lines and several General Purpose Input/Output pins (GPIOs).

Although I^2C is currently dominant in CubeSats, many developers experience in-orbit issues with this bus (see Chapter 2). Specifically, in-orbit bus lockups of the I^2C data bus, the large connector, lack of a clear standardized power bus distribution and protection and lack of a fixed pin allocation were identified as key issues. The Delfi-C^3 CubeSat suffered from a high bit-error rate and bus lock-ups [3] with I^2C in-orbit. From these lessons learned, it can be concluded that the theoretical behavior of a data bus does not always apply in practice.

Another study proposes a split data and power interface using daisy-chained connections [4] and call this the CubeSat Next Generation Bus (CNGB). For the data interface, the CAN bus was chosen with, as main reasons, the high level of hardware supported features and extensive heritage in the automotive industry. Details on the trade-off are, however, not provided. The paper mentions extensibility to larger than 3U-CubeSats as one of the programmatic goals. The split data and power connectors in a daisy-chained configuration is far from a small and lean solution and would not be suitable for smaller CubeSats or PocketQubes.

For PocketQubes, the only existing standard is PQ60 [5]. This standard is more clearly defined than the PC/104 implementation on CubeSats. It defines the connector, the pin allocation and the printed circuit board outline. It supports several different power outputs, SPI and I^2C data interfaces and many GPIOs. On the other hand, it uses a proprietary connector which is limited in current (only 0.2 A per pin). The literature described above shows that most used and proposed electrical data busses are aimed at versatility, leaving a large design freedom to the subsystem developers. The disadvantage for these standards is that they do not guarantee compatibility and are far from optimal in terms of wiring harness. A lean standard with a minimum number of clearly defined interfaces would counter these issues, but the lack of design freedom require a careful trade-off of the data bus and architecture for power distribution.

Summarizing the findings, the following top level goals for electrical bus interfaces are established:

1. The interface is lean in volume and wiring harness.
2. The interface has a fixed data bus and power distribution.
3. The interface supports expected future performance demands.
4. The interface enables a high satellite power efficiency.
5. The interface is low in complexity.
6. The interface is expected to receive support in the community.
7. The interface is robust and reliable.

3.2. Data Bus Architecture & Candidates

Before selecting a data bus for an electrical interface standard, it is helpful to define a data bus architecture for a typical CubeSat or PocketQube.

3.2.1. Data Bus Architecture

For this study, it is assumed that both satellite form factors make use of a distributed computing architecture, in which each physical subsystem of the satellite has its own micro-controller (or processor) to manage the local functionality. Some physical subsystems have components for which a digital interface is required, such as temperature sensors and reaction wheels. When they are physically implemented on the same board, a local data bus can be used, which can be of different kind and/or network topology (e.g. SPI). A central On-board Computer (OBC) manages the satellite by commanding the distributed micro-controllers and acquiring (housekeeping) data. Very advanced concepts, for example fractionated spacecraft or decentralized real time operations without a master node (central OBC), is considered out of scope for this study. While these concepts may have potential in the future, it is unlikely that these would receive wide community support in the short term.

For CubeSats and PocketQubes it is expected that for housekeeping data and internal commands, a linear bus connected to all physical subsystems will suffice.

In a linear bus network topology, the same set of wires or lanes are used to connect multiple nodes on the bus together. This is different from a point-to-point bus, which can only connect two nodes together. A linear bus has the major advantage for very small satellites that the amount of wiring is limited when stacked connectors or some form of bus backbone is used. Secondly, the pin-out is fixed for all subsystems and the number of potential nodes is not constrained by the amount of wiring.

A higher data rate of a linear data bus will support modest payloads connected to the same bus, which maintains a simple architecture. A linear data bus is, however, limited in speed because of cumulative electrical capacitance on the bus when adding nodes and the increasing demands on all nodes in terms of clock frequency and data handling capacity. Sophisticated and demanding payloads such as optical instruments produce, besides some modest housekeeping data, large amounts of payload data which may need to be stored and sent to selected ground stations over a high-speed radio transmitter [6]. In a study on CubeSat science missions [7], it was found that high-speed radio links up to 100 Mbit/s are currently commercially available and being integrated in CubeSats. For these type of payloads it is expected that point-to-point busses will be required between the payload, potential data storage and a high-speed radio.

Wireless communication inside a CubeSat is not common (see Figure 2.6 in Chapter 2), but a few experiments have been performed. A wireless sun sensor [8] using a proprietary wireless standard has been demonstrated on Delfi-C^3. A custom optical variant of the CAN bus has even been demonstrated as main data bus [9]. The advantages of wireless communication become most apparent for sensors which are remote from the internal printed circuit board and could potentially be self-powered and thus completely wireless [10], e.g. sun sensors. Wiring, in this case, is typically a major burden. Whenever there is potential for these sensors to locally power themselves, wireless data busses may provide a great solution. In a previous study, Bluetooth 4.0 was evaluated as one of the current best options [11]. For data communication between the main subsystems, where a wired electrical interface is required for electrical power distribution, the potential reduction in wiring harness is limited while complexity would increase.

Figure 3.1 shows the proposed data bus architecture, which is considered to be appropriate to fulfill the requirements of many CubeSat and PocketQube missions in the near and long term future. All subsystems and payloads are connected to a linear housekeeping bus which is mastered by the On-board Computer (OBC). Low-speed payloads can use this bus for payload data as well. Sophisticated payloads, together with data storage and a high speed transmitter, should use point-to-point busses to make a high data throughput possible while relieving the OBC for its critical tasks. Remote self-powered wireless sensors connect to the OBC and/or Attitude Determination and Control System (ADCS) through either wireless links or dedicated local data bus branches. It should be noted that individual CubeSats and PocketQubes can deviate in terms of number and types of physical subsystems. The centralized concept, where the OBC manages the satellite as a master device is a starting point for further analysis. In this architectural concept the OBC can still be physically allocated anywhere or combined with other subsystems.

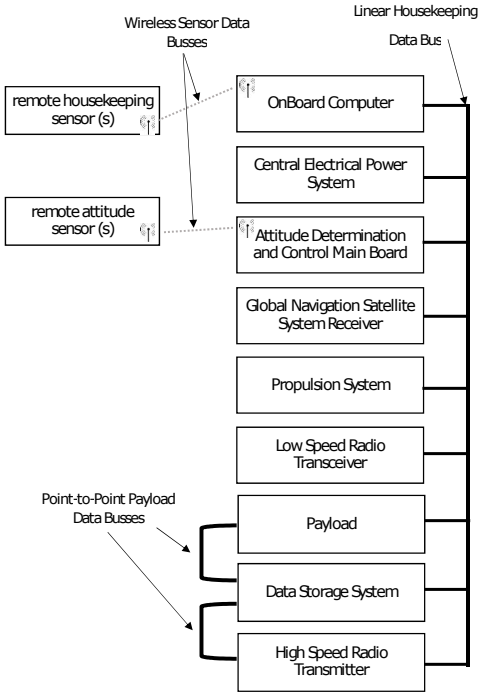


Figure 3.1: Proposed data bus architecture (example).

3.2.2. Linear Housekeeping Data Bus Candidates

The primary focus for this study is on data busses which are specified by a physical layer (ISO layer 1) [12]. As the number of existing busses and their variants is large, first a selection has been applied based on the high level goals described in section 3.1.1. Next to these goals, only data busses which are widely applied in terrestrial environments are considered. CubeSats and PocketQubes benefit from the associated wide availability of commercial integrated circuits, test equipment, documentation and user support for these data busses. The candidates considered for the linear housekeeping data bus are: Inter-Integrated Circuit (I^2C), differential I^2C, Controller Area Network (CAN) and Recommend Standard 485 (RS-485).

I^2C is a single ended synchronous bus: it has clock and data lines [13]. The lines are actively pulled high by a resistor (typically 4.7 kΩ) and have to be pulled low by its controller for communication. When applied in a small satellite, bus buffers need to be added to be able to isolate unpowered subsystems from the main data bus.

I^2C can be made differential by replacing the bus buffers by a dedicated differential driver [14], which yields four lines in total. As this is an easy-to-implement feature that slightly deviates from the standard while improving the robustness of the bus, this variant is added even though it is not widely implemented yet.

CAN is an asynchronous differential data bus developed for the automotive in-

dustry [15]. Some micro-controllers include a CAN controller, but most require an external controller connected to a local data bus that is supported internally by the micro-controller (e.g. SPI). An external differential driver is required in both cases.

RS-485 is an asynchronous differential data bus. It uses the Universal Asynchronous Receiver Transmitter (UART) that can be found on almost every micro-controller [16]. A dedicated external differential driver is required to make a RS-485 bus. This bus is the only one of the four options which is only specified on the physical layer and not on the higher OSI (Open System Interconnection) layers.

3.3. Data Bus Trade-off Process and Test Set-up

This section describes the trade-off method, criteria, test setup and community survey input.

3.3.1. Trade-off Method

Trade-offs with multi-disciplinary criteria are sensitive to errors and subjective scoring and weighting. Furthermore, a typical pitfall is to assign scores relative to the option space rather than the overall project or system scope. For example: a component trade-off leads to the discovery of several options ranging from €2 to €20. If the option space would be used to define a linear scoring range from 1 to 10, the individual score would be equal to the component cost divided by €2. The cheapest option receives a score of 1 and the most expensive receives a score of 10. This may make sense for a €100 mobile phone, but not for a 100 k€ satellite project.

Methods dealing with some of the sensitivities of trade-offs exist, such as the well-established Analytic Hierarchy Process (AHP) [17]. This method provides a structured approach to derive criteria, relative weighting of these criteria and the grading of all options for each criterion. Saaty, however, also states that the interpretation of an option within a certain criteria, even if these itself are objective facts, is always subjective [17]. The AHP method uses pair-wise comparisons between criteria and options to simplify the choices for the user. The fundamental scale used for these comparisons is presented in Table 3.1. Each pair-wise comparison enters together with its reciprocal in a n x n matrix, where n is the number of options. When the table is filled, the normalized eigenvector of the matrix is calculated to provide the resulting priorities (weights) for the options. Different weights to multi-disciplinary criteria can lead to the most acceptable compromise between different subjective perspectives. While the weighting between criteria is, per definition, subjective and requires only high-level expertise, grading can be based on facts and requires more detailed insight into the topic. For the trade-off of the housekeeping data bus it was chosen to derive the criteria and set-up a grading table for the options per criterion, iterated by several staff members at TU Delft with data bus experience. For the weighting between all criteria and the scoring of some criteria, the community was involved in the AHP using a questionnaire as elaborated in Section 3.3.5.

Table 3.1: Fundamental Scale for Pairwise Comparisons in AHP [17].

Importance		Explanation
1	equal	Two elements contribute equally to the objective.
3	moderate	Experience and judgement moderately favor one element over another.
5	strong	Experience and judgement strongly favor one element over another.
7	very strong	One element is favored very strongly over another, its dominance is demonstrated in practice.
9	extreme	The evidence favoring one element over another is of the highest possible order of affirmation.

3.3.2. Derivation of Trade-off Criteria

In Figure 3.2 a first derivation of trade-off criteria is presented, which comes from the design targets described in section 3.1.1.

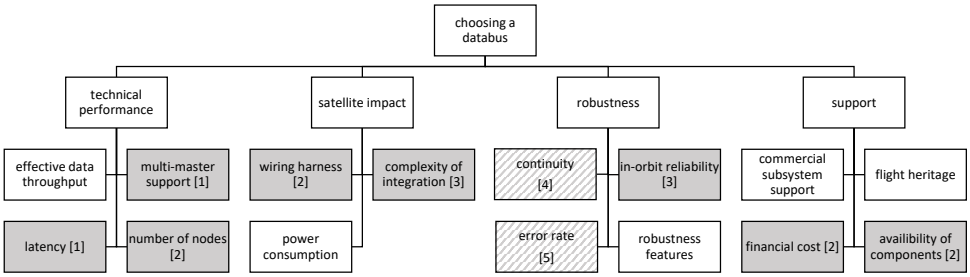


Figure 3.2: Derivation tree for criteria (grey/patterned boxes are omitted after theoretical/practical analysis).

Some of the identified criteria are omitted after theoretical analysis. These boxes are marked solid grey in Figure 3.2 and the numbers between brackets refer to the following reasons:

1. These criteria are not considered to be significant. A housekeeping data acquisition and commanding cycle in the order of 1 – 10 Hz, managed by the On-board Computer as master, is a typical approach [18] that works very well and does not require low latency or multi-master support.

2. The difference between the data bus options for these criteria are considered to be too small or out of scope. RS-485 supports 32 nodes and the others even a few hundred. RS-485, CAN and I^2C require two wires and dI^2C just four. For all busses, the required integrated circuits are widely available from different manufacturers and are all very low in cost (a few € or US$).

3. There is no good metric or data available for these criteria. For complexity of integration, there is too limited community experience for RS-485, dI^2C

and CAN to aggregate subjective input. Sufficient statistical input for these busses is also missing for in-orbit reliability, which would give I^2C an unfair disadvantage.

As a next step, initial laboratory tests (see Section 3.3.4) have been performed to discover if the derived criteria can deliver appropriate results which can be used for comparison with a reasonable amount of effort. This led to a further reduction in criteria after practical analysis, for which the boxes are patterned grey for the following reasons:

4. Continuity as criterion refers to the ability of the data bus to operate continuously with bus lockups or other events which cause temporary unavailability of the bus. The chosen metric for this is the amount of disruptive events per time unit, in which less than once per 24 hours would receive the highest grade. In the initial tests, all four data busses did not show any such disruption, even when subjected to electromagnetic interfere (see next point).

5. Error rate as criterion refers to the number of (bit) errors per number of transactions or bits. The chosen metric was the Packet Error Rate (PER) which could be discovered by a check of the Cyclic Redundancy Check (CRC) in each transaction. A packet error would indicate one or more bit errors within the transaction. A PER of less than one-in-a-thousand would receive the highest grade. All four data busses were tested for about 30,000 transactions each. In ambient conditions none of them showed packet errors. Tests have also been performed at high computational load on the micro-controllers (i.e continuously calculating the value of pi) while the micro-controllers receive interrupts op to 1000 Hz from an external generator connected to one of the I/O which was given a high interrupt priority. In all those tests, no packet errors have been detected. Finally, tests have been performed by injecting simulated Electro-Magnetic Interference (EMI). First by a direct injection of white Gaussian noise on the bus lines with capacitive coupling and a signal generator. All data busses withstood a noise injection up to 0.8 V RMS without any packet error, but it must be noted that the peak-to-peak voltage levels generated by the used signal generator are, in this case, already beyond 10 V. This is significantly higher than the signal reference level of 3.3 V used by all data busses and beyond the electrical specification of their integrated circuits. Only at even higher noise levels, the busses showed packet errors and lockups. Lab experiments with a spare model of Delfi-C^3 and subsystems of Delfi-n3Xt (specifically the reaction wheels and magnetorquers) showed noise levels below 1 V. These satellites are not representative for all CubeSats and PocketQubes, but show that a sample selection of a few subsystems is not appropriate to identify EMI sources which do results in disruptions and communication errors. Other tests were performed to simulate power transients on lines with switching currents of several amperes, including in-rush currents of several tens of amperes. In all cases, there were no packet errors discovered. After several experiments it became clear that all busses are resilient to

a significant amount of noise. Still, there is insufficient knowledge of EMI levels, characteristics and test methods which would be appropriate to simulate a wide scale of PocketQube and CubeSat configurations including more "exotic" components (e.g. pulsed plasma thrusters) within a reasonable amount of effort. It is therefore decided to omit test-based inputs for error rates and only focus on inherent robustness properties of the data busses themselves.

The experiences with the test setup are not in line with the in-orbit experiences with the I^2C data bus as described in Section 2.3. During the development of the test setup and even the initial EMI testing, bus lockups and significant errors appeared on all tested busses. This resulted in the discovery of several flaws in the software drivers of the test setup which have been corrected appropriately. The test setup used for this paper is based on all the same micro-controllers and the software is extensively debugged, which is different from Delfi-C^3 which comprised several different micro-controllers and was launched with known (and accepted) frequent packet losses and bus lockups. This is potentially also the case for other flown CubeSats. In the specific example of Delfi-C^3, it was found that the clock speed of the micro-controllers, the I^2C software drivers and differences between the I^2C hardware drivers within the micro-controllers have caused disruption and significant error rates [3]. It is expected that I^2C problems on Delfi-C^3 could have been solved before launch but would have required extensive testing, debugging of software and even changes to the hardware. The experiences show that in-orbit experiences cannot directly be projected to the intrinsic reliability of a data bus and that a fair comparison on reliability can only be performed if the both hardware and software are extensively tested and corrected for development errors and/or inadequate choices for relevant components. The remaining criteria are worked out further and some sub-criteria are added. Figure 3.3 presents the final trade-off criteria tree for choosing a data bus.

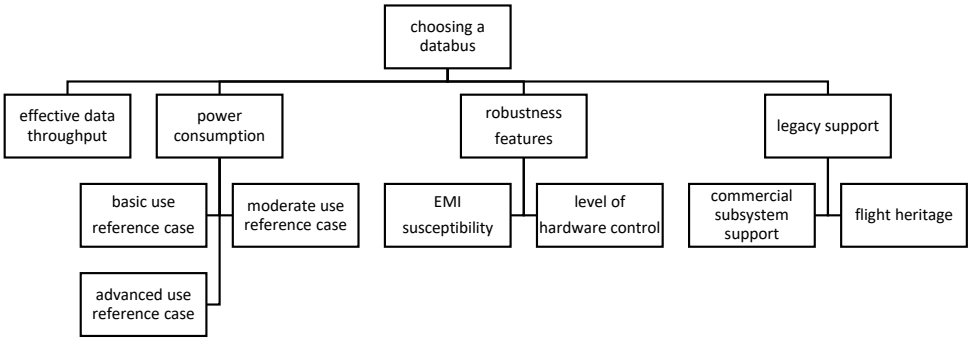


Figure 3.3: Final criteria tree for trade-off.

Effective data throughput refers to the maximum amount of data which can be transferred over the bus from the master (OBC) to the slaves and back. It is the sum of all message content over the bus, excluding addressing, protocol overhead and timing delays.

The power consumption of the linear data bus is dependent on the number of nodes and the data throughput. As this may vary between missions, three reference use cases for the linear data bus have been defined:

- Basic: a satellite with 5 subsystem nodes with a data and command cycle of 1 Hz. Payload could be a very low data rate sensor or a technology demonstration of (part of) a subsystem.
- Moderate: a satellite with 9 subsystem nodes with a data and command cycle of 1 Hz. Payload could be similar to the basic case or could be sophisticated using dedicated point-to-point data bus(es) as depicted in Figure 3.1.
- Advanced: a satellite with 9 subsystem nodes at a relatively high data rate compared to the basic and moderate case. The high data rate can be attributed to a significantly higher data and command cycle and/or a payload with moderate data rate which does not yet justify a dedicated point-to-point data bus. The effective data rate is fixed to approximately 250 kbit/s for this case, which was expected to be supported by the four chosen options.

The robustness features are EMI susceptibility and level of hardware control. The best attribute to judge EMI susceptibility on, based on the four options, is the difference between non-differential (I^2C) and differential (dI^2C, RS-485 and CAN), where the latter is generally less susceptible due to common mode noise rejection. Testing under normal conditions did not show any errors including for regular I^2C. More intense EMI environments are unknown, so there is no quantitative metric based on value input possible for this criteria. It is therefore chosen to ask the community on their judgement, using the fundamental scale of AHP to determine the relative grades. For the level of hardware control, pairwise comparisons between three levels have been used:

- Large part of the data protocol and potential error detection and failure handling needs to be implemented in the software (RS-485).
- A hardware controller for the full data protocol, but where the potential error detection and failure handling needs to be implemented in the software (I^2C & dI^2C).
- A hardware controller for the data protocol including internal error detection, correction and failure handling (CAN).

RS-485 requires the full data protocol and any software error detection and correction to be fully implemented in software. The UART and the differential driver only provides the physical layer. This means that the micro-controller needs to allocate relatively the highest amount of resources to the data bus and potential software bugs or interrupt/state control within the micro-controller could more easily lead to anomalies on the data bus compared to hardware control. I^2C and dI^2C do have the data protocol defined and implemented in the hardware controller. This will offload the micro-controller and is less prone to software bugs. CAN even

has error detection and correction included in the hardware controller, which would make it most robust in this respect. However, the statements above are only true if the hardware controller has no flaws in the state-machine. Practical experience with I^2C shows that this is not always the case [3] and the high number of bus lockups experienced by developers in orbit (Section 2.3.1) may be an indication of a larger problem. Given the high degree of subjectivity in this matter, grading for this criterion is again based on the community judgement in pair-wise comparisons.

Finally, the legacy support of the data busses is taken into account. One sub-criterion is the commercial subsystem support. The rationale is that, of all available commercial subsystems, one can more easily and quickly adopt the wiring interface if the data bus is already supported. Alternatively, one can use a relatively simple interface-to-interface connector for the new proposed electrical interface standard compared to a situation where the subsystem does not yet support this data bus. The second sub-criterion is the flight heritage, which is based on the results of a survey performed on CubeSats (Section 2.3.1). Both criteria are value based, but in terms of relative grades they do not have a direct technical impact on the satellite such as the effective data rate or power consumption. Therefore the community is asked to define the grading range for each.

3.3.3. Grading for Final Criteria

As next step, the grading is determined for the trade-off, which is presented in Table 3.2. The grade ranges for the criteria using quantitative input are based on internal experience as well as studies of worldwide CubeSats [19].

Table 3.2: Grading table for linear housekeeping data bus.

Criterion	Grade	Input
Data Throughput	$\frac{D}{1000kbit/s}$	D is the effective data throughput.
Power Consumption	$1 - \frac{P}{T}$	P is the total power consumption. T is the threshold (if $P > T \rightarrow$ reject) for PocketQube/CubeSat: $T_{basic} = 50mW/200mW$ $T_{mod.} = 100mW/400mW$ $T_{adv.} = 200mW/800mW$
Robustness Features	Fully AHP survey based, see Section 3.3.5.	
Legacy Support	$\frac{1 + (S-1) \cdot I}{S+1}$	S is the AHP scale factor (see Table 3.1). I is the implementation rate.

The AHP method uses normalized grades and weights in which the individual grades for the options and the weights of the criteria need to add up to one. Therefore, some of the grades from Table 3.2 need to be normalized before entering the next step of the trade-off. It also should be noted that community experience for CubeSats is also considered as input for PocketQubes as it involves flight heritage on very small satellites and public documentation on implementation lessons learned.

3.3.4. Housekeeping Data Bus Comparative Test Setup

This section describes the final test setup for the input for grading effective data throughput and power consumption.

The test setup comprises up to nine Texas Instrument MSP432 micro-controller development boards. The MSP432 is a modern micro-controller which is chosen as the default controller for the Delfi-PQ PocketQube of TU Delft due to its low power over computational load ratio. The data bus specific hardware is placed on daughter boards which can be stacked on top of the development boards. A ribbon cable connects all boards. The power consumption is measured at the input power which is run to all boards by means of a high precision current meter. Before each test, the power consumption of each development board is measured without the daughter boards. This value is subtracted from the measured power during the data bus tests. The complete setup is shown in Figure 3.4.

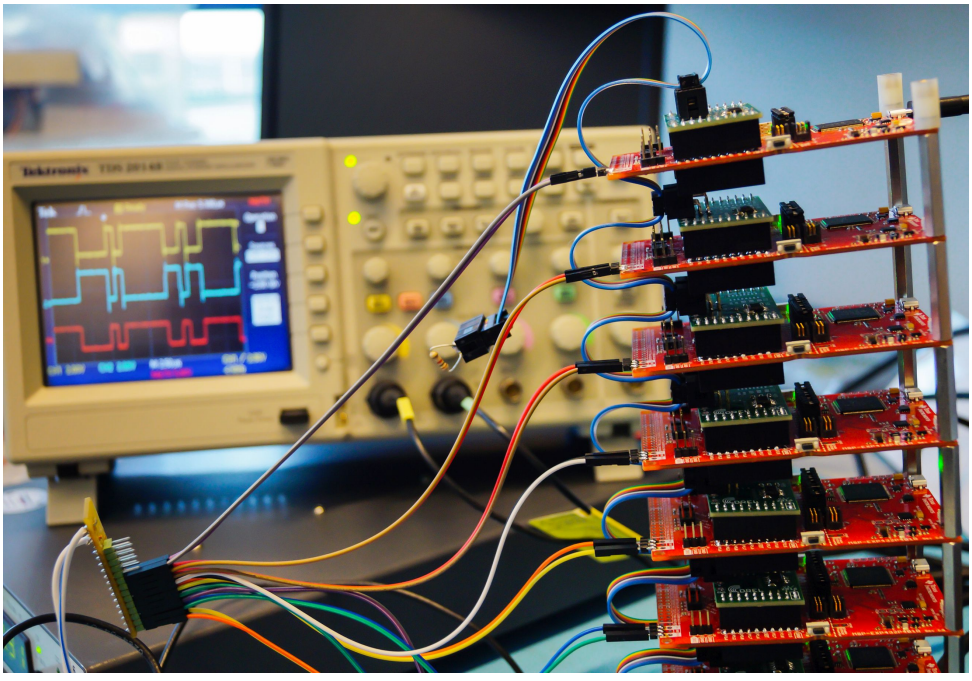


Figure 3.4: Test setup for data bus characterization.

For I^2C, the dedicated internal controller on the MSP432 is used and a data bus

buffer is added per board. The circuit is represented in Figure 3.5. For differential I^2C, the data bus buffer is replaced by a dedicated differential driver as shown in Figure 3.6. For RS-485, the UART of the MSP432 is used and a dedicated differential driver is added to the UART as shown in Figure 3.7. For CAN, both an external controller and a driver are required as shown in Figure 3.8. CAN is the only data bus under consideration which is not supported with an internal controller on-board the micro-controller chip. It has to be noted that there are some micro-controllers available with internal CAN controllers. This may positively influence the power consumption, but this will limit the choice of micro-controllers severely and may require major adaptations of existing subsystem designs. For all data busses, a list of potential components are selected which operate at 3.3 V level. From this list, the ones with the lowest power consumption according to the manufacturer specification are selected out of a list of options from different manufacturers. For all busses, bias and termination resistors are chosen following the recommended specification to ensure optimal behavior and noise rejection.

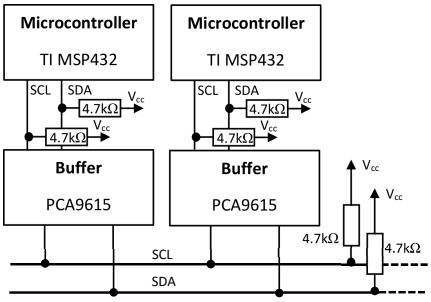


Figure 3.5: I^2C circuit.

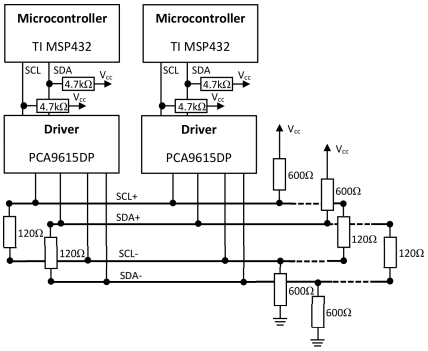


Figure 3.6: dI^2C circuit.

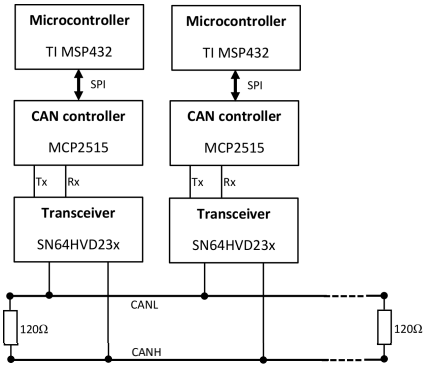


Figure 3.7: CAN circuit.

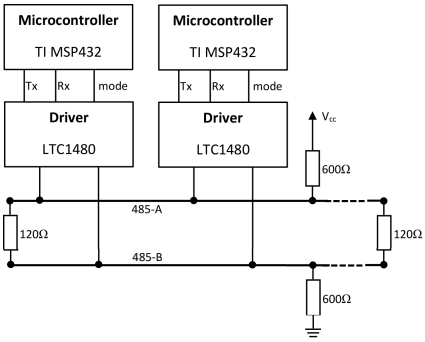


Figure 3.8: RS-485 circuit.

For testing the power consumption and throughput efficiency, a reference case communication scenario has been established in Table 3.3 which is based on both the architecture and example provided in Figure 3.1. It is assumed that this stan-

dard communication set is cyclic at 1 Hz. The data packet size are based on experience with Delfi satellites and commercial CubeSat hardware. Large packets, not supported by a data bus (e.g. CAN), will be broken up in sequential packets.

Table 3.3: Reference communication set for linear housekeeping data bus.

Source Node	Recipient Node	Size [bytes]
1. OBC	3. EPS	2
3. EPS	1. OBC	30
1. OBC	4. ADCS	2
4. ADCS	1. OBC	120
1. OBC	6. GNSS	2
6. GNSS	1. OBC	30
1. OBC	7. propulsion	2
7. propulsion	1. OBC	10
2. OBC	2. H/K radio	2
2. H/K radio	1. OBC	10
1. OBC	5. payload	2
5. payload	1. OBC	10
1. OBC	9. data stor.	2
9. data stor.	1. OBC	10
1. OBC	8. P/L radio	2
8. P/L radio	1. OBC	10
1. OBC	9. data	250
1. OBC	2. H/K radio	250
	Total of node 1-5, 9 packets:	428
	Total of node 1-9, 18 packets:	746

While the reference communication set is a realistic representation of the architecture and subsystem structure provided in Figure 3.1, it does not apply for satellites with modest payloads that may not require dedicated payload data busses. Also, the frequency of 1 Hz is arbitrary and can be higher or lower depending on the specific needs of the mission. To determine the maximum effective throughput of the data bus, the set in Table 3.3 is simply looped continuously. For satellites with payloads using relatively large data packets, the average overhead may decrease and thus the effective throughput maybe higher. It is, however, expected that the variations for different scenarios will not lead to very large deviations in the outcome and will be even more marginal, in a relative sense, between data busses.

3.3.5. AHP Questionnaire for Community Input

A questionnaire has been set up and sent in March 2017 to 36 and 453 members of the PocketQube and CubeSat community respectively. It has been decided to keep these communities separate, as the characteristics of these two different form factors are very different (in terms of volume, power, sophistication of payloads, flight heritage, etcetera). The questionnaire had a response of 34 participants from the CubeSat community, representing 30 different developers from around the world.

Likewise, there were 15 participants representing 10 different developers from the PocketQube community. All questions provide input for the mutual weighting of sub-criteria followed by the main criteria in pair-wise comparisons using the AHP scale (see Table 3.1). For example, it is asked: *What do you consider to be more important: effective data throughput or power consumption, and to which extend (equal, moderate, strong, very strong or extreme)?*. The full questionnaire and the anonymized answers can be found in the 4TU repository [20]. All criteria weights and the specified grades are determined using the questionnaire input and an Excel-based tool [21] that calculates the AHP output.

3.4. Trade-off Results on Data Busses

3.4.1. Power Consumption

The test results on the power consumption of the data busses are presented in Figure 3.9 and Figure 3.10. The graphs shows the total power consumption of each data bus for the number of bus nodes attached. The standard deviation between the four independent test runs for all test points is 6.5 mW. The confidence interval CI can be determined by:

$$CI = \bar{x} \pm z^* \frac{\sigma}{\sqrt{n}} \tag{3.1}$$

where:
$\bar{x}$ = mean
z^* = confidence interval index
σ = standard deviation
n = number of measurements.

The 95% confidence interval ($z^* = 1.96$) for the four test runs ($n = 4$) is $\pm$6.4 mW for the data presented in Figure 3.9 and Figure 3.10.

For the trade-off, the input data are taken from the 5 node and 9 node values in Figure 3.9 and the 9 node values from 3.10 as explained by the use cases in Section 3.3.2. The values are presented in Table 3.4. For the advanced use case at 250 kbit/s, the CAN power consumption is extrapolated from its maximum data rate of 136 kbit/s and idle consumption. The power consumption for 9 nodes at the maximum data throughput, which is not used in the trade-off, is also provided.

Table 3.4: Power consumption for the trade-off used reference cases.

	Power Consumption [mW]			
Use Case	**I²C**	**dI²C**	**CAN**	**RS-485**
basic	52	36	139	9
moderate	95	63	268	11
advanced	141	153	362	59
maximum	139	154	318	108

The grades are calculated by entering the data from Table 3.4 into the grade

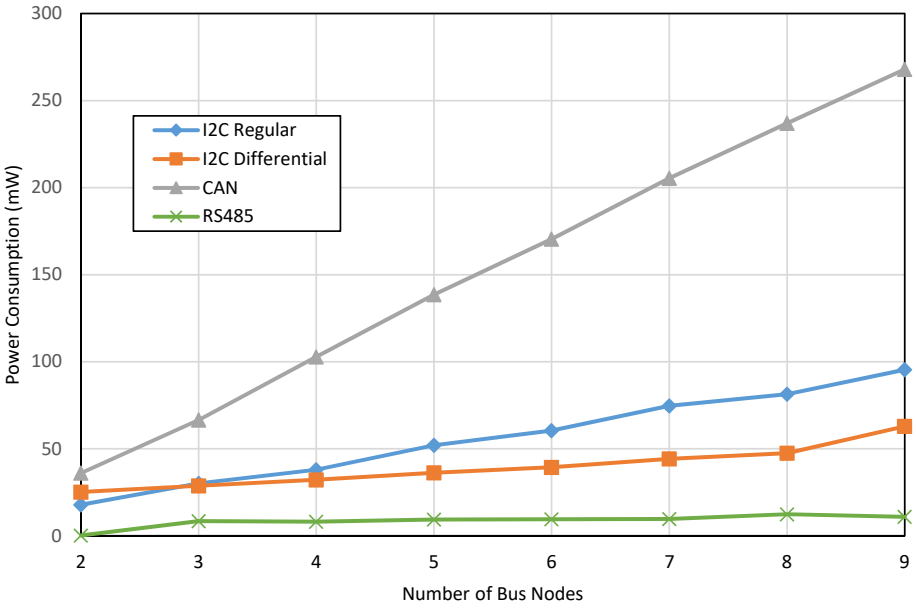


Figure 3.9: Power consumption for one communication set per second.

equation in Table 3.2. As a next step, the grades have been normalized to the sum of one (as required by AHP) and are subsequently multiplied by the calculated relative weights per participant following from the community survey. This yields individual priorities (grades) for the criterion of power consumption. For CubeSats, the mean weights of all participants are 0.29 for the basic, 0.31 for the moderate and 0.40 for the advanced use reference case. For the PocketQubes these are 0.42, 0.16 and 0.42 respectively. The priorities are presented in Figure 3.11 which shows a box-plot for the spread of individual priorities. The end of the legs show the minimum and maximum, the end of the boxes show the first and third quartile of all participants and the line in the middle shows the median. Additionally, the cross shows the mean of all participants and the dot shows the relative number of participants for which the specific data bus received the highest priority.

From Figure 3.11 it can be concluded that for PocketQubes, RS-485 has a clear advantage over the other busses. CAN, on the other end, does not meet the rejection threshold and should therefore be omitted as option for PocketQubes. Because of limitations of the AHP method, it is still included in the final trade-off with the grade for this criterion set to zero. For CubeSats, the spread of priorities for this criterion is significantly less, which can be explained by the higher reference power levels as presented in Table 3.2.

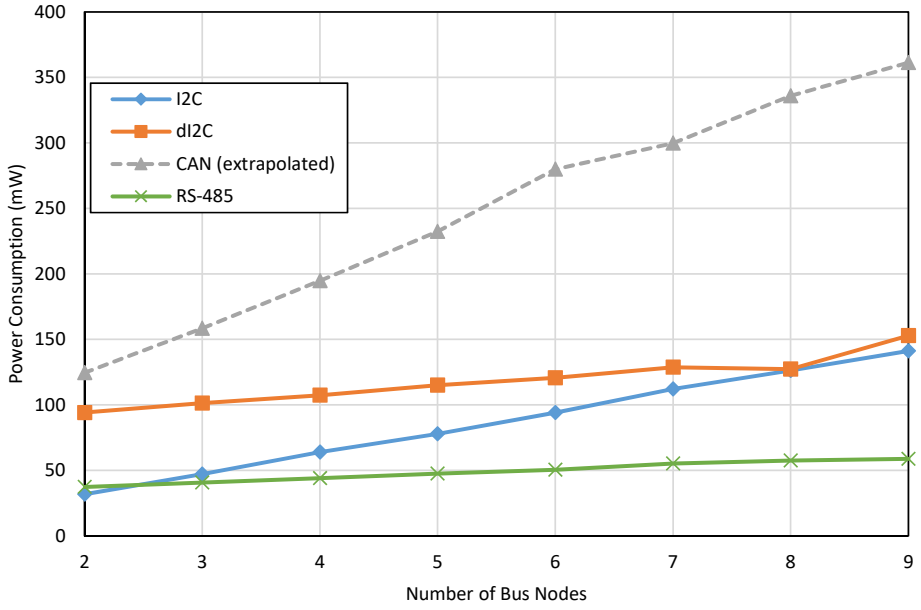


Figure 3.10: Power consumption for 250 kbit/s.

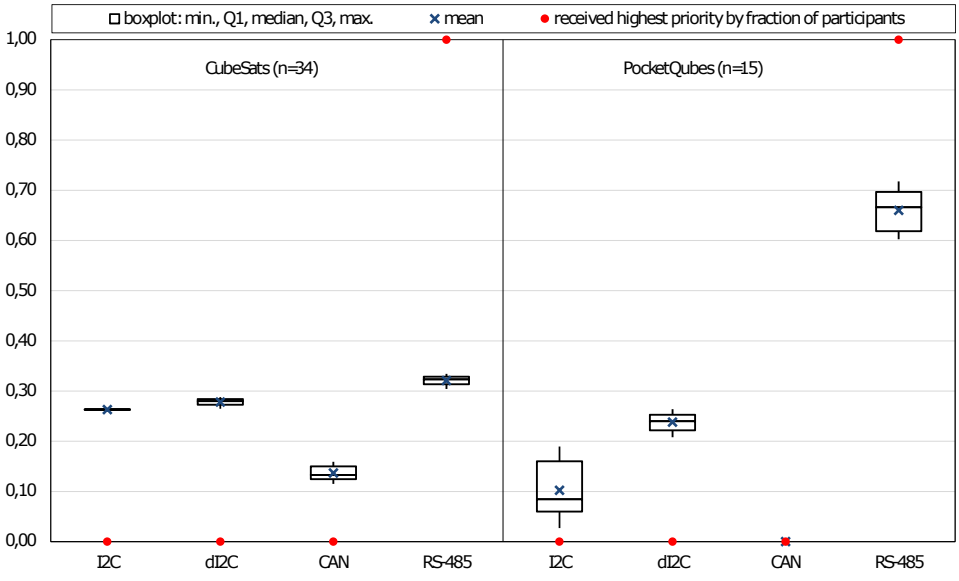


Figure 3.11: AHP priorities on power consumption.

3.4.2. Effective Data Throughput

Table 3.5 provides the effective data throughput at the advanced reference case which is used for input of the trade-off. The initial grades based on Table 3.2 are normalized to calculate the AHP priority.

Table 3.5: Effective data throughput at advanced reference case

Data Bus	Controller Baud Rate	Expected Data Efficiency	Effective Data Throughput	True Data Efficiency	AHP Priority
I^2C	400 kHz	80%	248 kbit/s	62%	0.20
dI^2C	400 kHz	80%	258 kbit/s	65%	0.21
CAN	1 MHz	51%	136 kbit/s	14%	0.11
RS-485	1 MHz	79%	600 kbit/s	60%	0.48

I^2C, dI^2C and RS-485 have all a theoretical calculated data efficiency of about 80% for the communication set in Table 3.3. The measured efficiencies are lower, which can be attributed to the latencies of about 20% of total transaction time within the micro-controller of handling the data.

One of the reasons for the relatively low effective data throughput and also low data efficiency of CAN can be found in the protocol overhead. A CAN frame with the maximum of 64 bits of message content is, in total, 114 bits (for the base frame format) including protocol overhead, so the efficiency is at best 56%. For a small 16-bit message, the total CAN frame is 66 bits, yielding an efficiency of 24%. Due to Non-Return-to-Zero (NRZ) encoding, bit stuffing is needed, which reduces efficiency up to 20%. The expected data rate in Table 3.5 is based on the communication set in Table 3.3 and 10% bit stuffing. Including the probable latency factor of the micro-controller, one would still expect an efficiency of approximately 40%. The best explanation for the gap between theory and test results are the latencies caused by the additional SPI interface between the micro-controller and the CAN controller. There is thus a potential gain in effective data throughput if internal controllers are used. The sensitivity of final trade-off for a theoretical improvement up to 400 kbit/s for CAN is investigated in Section 3.4.5.

3.4.3. Robustness Features

For determining the priorities on the main criterion 'robustness features', the AHP community survey is used for prioritization of the sub-criteria. This is explained in Section 3.3.1. The priorities are shown in Figure 3.12. CAN receives the highest priorities since it is a differential bus and has a high degree of hardware control. The mean relative weighting between the two sub-criteria is almost equal for CubeSats and PocketQubes, leading to a balance between I^2C and RS-485 and a slight advantage for dI^2C.

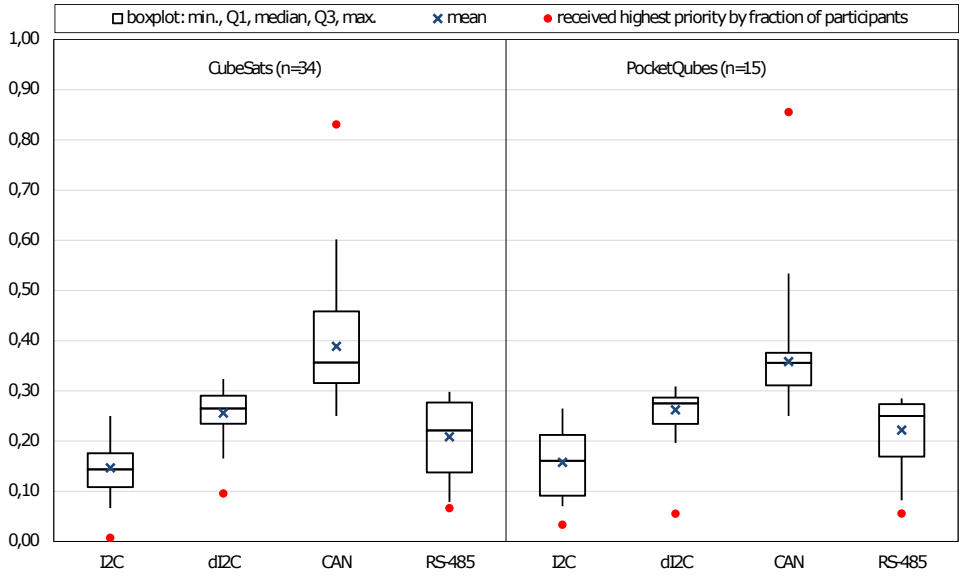


Figure 3.12: AHP priorities on robustness features.

3.4.4. Legacy Support

For the grading of legacy support using the equation in Table 3.6 for the sub-criterion 'flight heritage', I is the implementation rate on CubeSats from survey (Section 2.3.1). For the sub-criterion 'COTS subsystem support', I is the implementation rate on commercial CubeSat or PocketQube subsystems. For CubeSats, a wide survey of the market has been performed within this study with a large variety of commercial suppliers and (for each supplier) different subsystems. In total 56 different main physical subsystems coming from 23 different manufacturers have been selected. For PocketQubes, only 3 commercial systems were found. This is very low, making this a sensitive input for which the impact on the final result will be checked. For both implementation rates I, dI^{2}C receives 50% of result for I^{2}C support and whenever UART is mentioned instead of RS-485 explicitly, this is counted for 50% as well. The rationale is that the change from I^{2}C to dI^{2}C and generic UART to RS-485 require small modifications for which a major part of the legacy support is maintained.

For this criterion, the grade input data is scaled to the AHP range as determined from the survey (see Table 3.2). The priorities for the legacy support are provided in Figure 3.13. I^{2}C receives the highest priorities, which can be explained by the input data. However, the levels of priorities are reduced in range compared to the input values as for both sub-criteria and both satellites form factors, the mean importance is rated moderate to strong. Some participants have given equal priority to each level of support, which is the reason that I^{2}C does not score 100% of the received highest priorities.

Table 3.6: Grade input data for data bus legacy support.

Data Bus	CubeSat Flight Heritage (n=56)	CubeSat Commercial Support (n=52)	PocketQube Commercial Support (n=3)
I^2C	78%	40%	60%
dI^2C	39%	20%	30%
CAN	5%	20%	0%
RS-485	4%	20%	10%

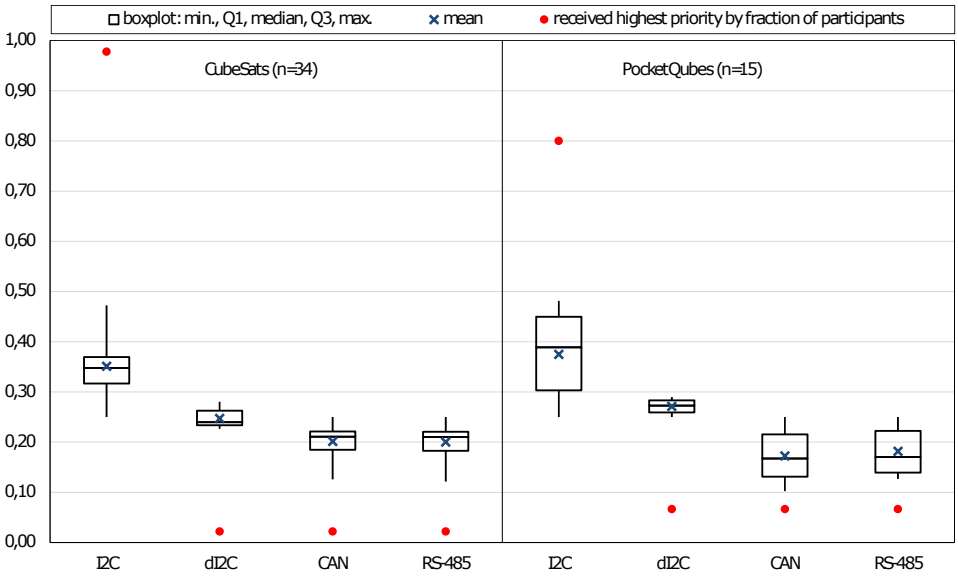


Figure 3.13: AHP priorities on legacy support.

3.4.5. Final Trade-Off

Finally, the weights between the four main criteria are determined using the AHP community survey and provided in Figure 3.14. The relative priority of each criterion is multiplied by its relative weight and summed for each option, leading to the final priorities as provided in Figure 3.15.

For PocketQubes, RS-485 received the highest priority for 12 out of 14 participants. This is explained by the high relative weight for power consumption in combination with the high relative priority on this criterion for RS-485.

For CubeSats, RS-485 also received the majority of highest priorities (19/34), followed by CAN (11/34). For CubeSats the criterion 'robustness features' received a high weight, which is in favor of CAN. Still, the combined weights on effective data throughput and power consumption and the relative good performance of RS-485 on these aspects swings the trade-off for many participants towards this data bus.

As mentioned in Section 3.4.4, the trade-off is potentially sensitive to the limited

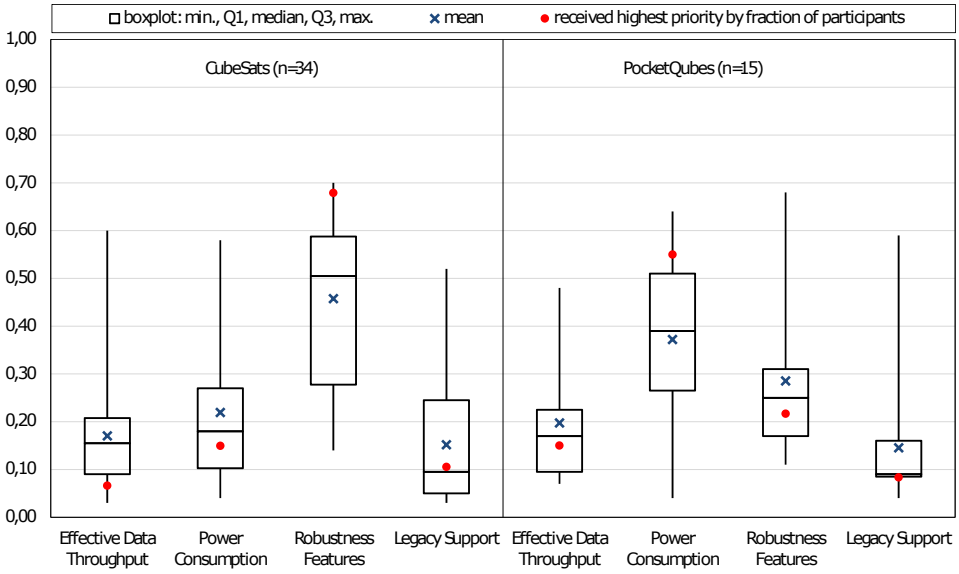


Figure 3.14: AHP weights of main criteria.

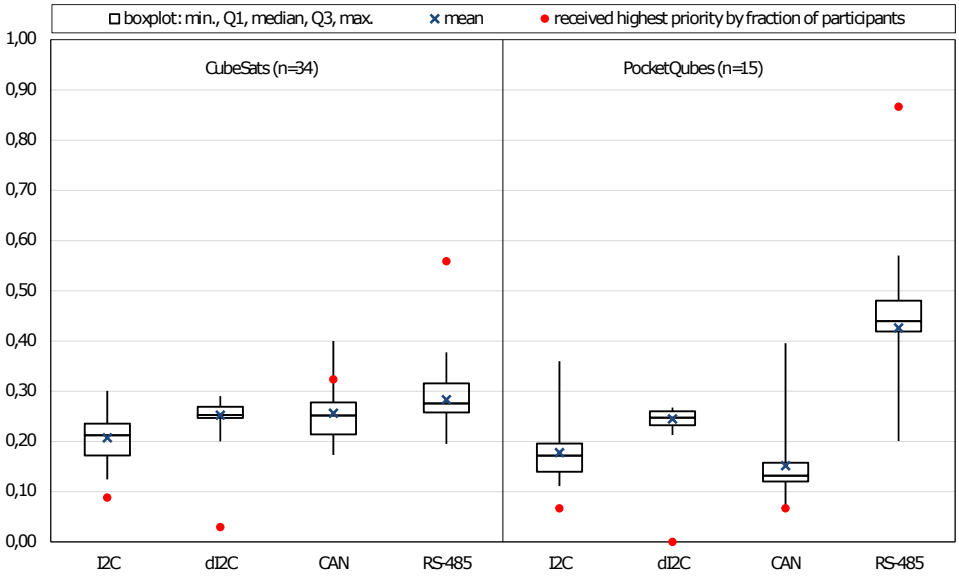


Figure 3.15: Final AHP priorities.

available commercial subsystems for PocketQubes for this study. If the sub-criterion would be omitted, the final priorities only changes slightly in favor of CAN and RS-485 while the distribution of highest priorities over the data bus options remain the

same.

As mentioned in Section 3.4.2, the effective data throughput of CAN in the test setup has been found to be significantly lower than expected. If this data rate would be improved to a theoretical data rate of 400 kbit/s, CAN would receive the highest priority by 13 out of 34 participants for CubeSats, while RS-485 would drop to 16 out of 34 participants. For PocketQubes, there is no effect on the final outcome of highest priorities.

3.5. Electrical Power Distribution Concept

In the Section 2.7 of Chapter 2 on the survey on CubeSat electrical interfaces, the conclusions and recommendations for a new interface standard were provided. Based on these conclusions, in combination with the (partially overlapping) design goals provided in Section 3.1.1, a power distribution architecture is designed for the proposed interface standard. The chosen philosophy is to keep the architecture lean and simple. A single-point-of-failure-free philosophy would required redundancy of all critical subsystems and interfaces and should allow for arbitration, which is in contrast with a lean and simple philosophy. For this reason, redundancy of subsystems is not included in the design, under the assumption that the impact on reliability of the satellite is acceptable. In Figure 3.16, a schematic overview of the power distribution is presented in which the unregulated battery bus is distributed via 4 or 8 configurable current protected switched outputs for PocketQubes and CubeSats respectively. Regulation occurs at the subsystems locally, which limits the number of required conversion steps and the impact of voltage drops over the wiring.

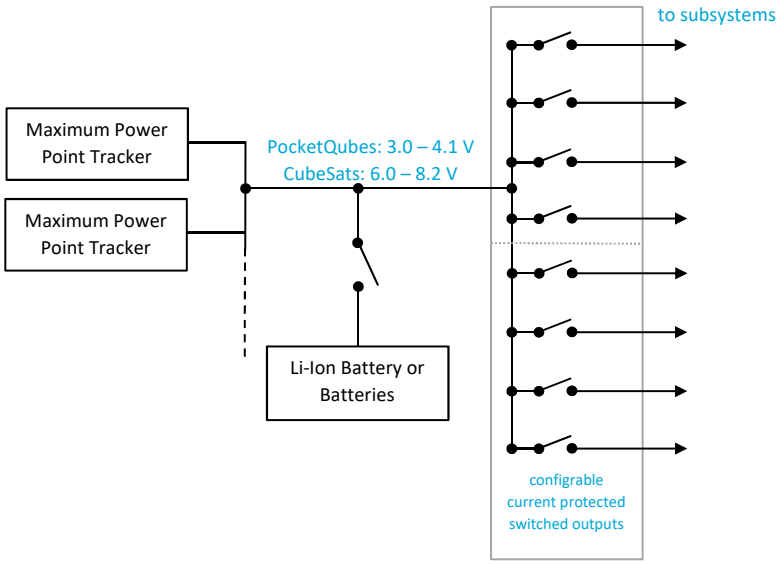


Figure 3.16: Power Distribution Schematic Overview.

Single Event Upset and software state errors can lock up a data bus or halt the operations of the OBC. In the current philosophy, subsystem redundancy as well as interface redundancy concepts are omitted. The central EPS can solve some issues with a power cycle of the full satellite, either at a default fixed interval (e.g. once per day) or when it does not receive for example a repeating synchronization message for a while from the OBC. Still, such methods do not mitigate all errors, for example on the central EPS itself. A reset line from the primary radio receiver to the EPS is recommended. The radio receiver should be able to decode a reset tele-command and pull the reset line high. The line is pulled low by a resistor and a decoupling capacitor near the input at the central EPS unit. At the central EPS, the power of the EPS micro controller and all distribution lines are taken down for a few seconds to enforce a true power cycle of all systems.

3.6. Proposed Electrical Interface Standard

Based on the trade-off results on the data bus and the analysis on the power distribution in Section 3.5 as well as the design targets stated in Section 3.1.1, the simplest solution for an electrical interface standard is defined and presented in Figure 3.17 and Table 3.7. For the PocketQube, a 9 pin interface connector is defined (the first 9 pins in the figure) and is called PQ9. For CubeSats, a 14 pin connector is defined in similar fashion, by adding 4 power distribution lines, and is called CS14. The first nine pins of CS14 are similar to PQ9, but due to the different voltage range, not identical. However, since the power distribution requires local regulation, it is very well possible that the local DC-DC converters can handle the entire input range from 3.0 to 8.2 V. This would create an opportunity to easily create a CubeSat version of a PocketQube system or to stack several of these PocketQube boards on a CubeSat motherboard.

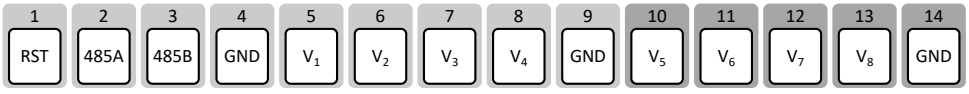


Figure 3.17: PQ9 (pin 1-9) and CS14 (pin 1-14) interface connector.

In Section 2.5.3, it is argued that a flex-rigid backbone in combination with side-mount connectors for wiring harness could limit the amount of board space required for the wiring harness, mainly when compared to PC/104. However, since the number of pins selected in this paper is very low, such a solution would not be optimal in terms of board space as the wire leads and insertion depth of side-mount connectors take more space. The final type of connectors and/or wiring harness chosen is a single row 2 mm pitched stack-able pin header connection. These connectors are low in cost, available in different stack heights, sold by different manufacturers and proven in space since they are very similar to the PC/104 connector. The mechanical outline of the printed circuit boards for PQ9 and CS14 are shown in Figures 3.18 and 3.19. A hardware example of PQ9 is provided in Figure 3.20. When comparing PQ9 to PQ60 it has about 15% of the pins and 30% of the connector footprint area. For CS14 compared to PC/104 this is 13% and 8% respectively.

Table 3.7: Pin allocation for PQ9/CS14 standard interface.

Pin	Signal	Allocation
1	RST	system reset line (60 Ω to gnd)
2	485-B	RS-485 inverting signal
3	485-A	RS-485 non-inverting signal
4	GND	ground
5	V1	recommended: OBC (PQ: + Radio)
6	V2	recommended: ADCS (PQ: + GNC)
7	V3	recommended: propulsion
8	V4	recommended: primary payload(s)
9	GND	ground
10	V5	recommended: radio
11	V6	recommended: GNC
12	V7	recommended: data storage & payload data transmitter
13	V8	recommended: secondary payload(s)
14	GND	ground

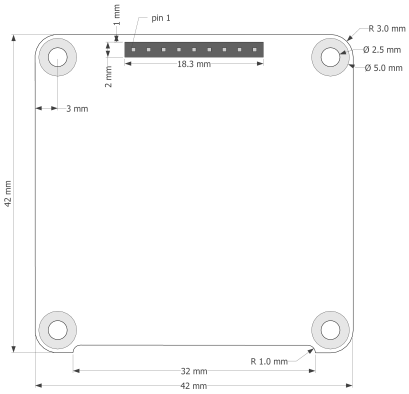


Figure 3.18: PQ9 PCB outline.

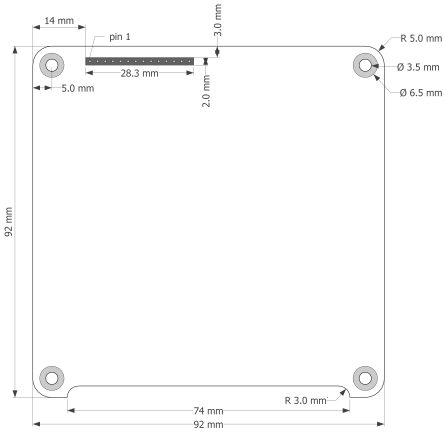


Figure 3.19: CS14 PCB outline.

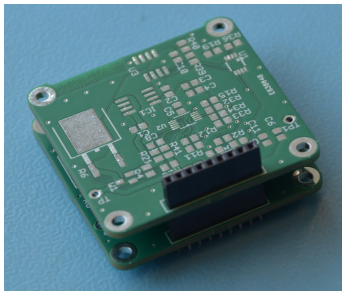


Figure 3.20: PQ9 PocketQube boards with stack-able pin connector.

3.7. RS-485 Data Protocol

This research and the data bus trade-off is limited to the physical architecture. It is obvious that in addition to any physical characterisation, data structures and software to implement an interface are crucial. Thus, this Section proposes also the data protocol for the RS-485 interface. RS-485 is the only data bus option considered which is only specified on the physical layer and not on the data layer [12]. However, for a proper interface standard, which shall yield compatibility between subsystems from different developers, the data layer will need to be specified. In this section, the data protocol as described in the PQ9 and CS14 interface standard [22] is provided. This data link protocol has a limited data overhead while already including integrity checks of the transaction and data which go beyond for example the I^2C standard for which this is not defined.

A byte is defined in Figure 3.21, in chronological order from left to right with the least significant data bit first. Each byte begins and ends with a start and stop bit, following the UART standard. When the address bit is set to one by any of the nodes, it causes an interrupt in the other nodes on the bus. Each node then checks if the address matches its own. If it is not addressed, the node can ignore subsequent bytes (address bits being zero). This additional address bit requires the micro-controller to support 9-bit UART with interrupt service. A small survey of several popular micro-controllers for CubeSats and PocketQubes shows that the majority of devices support this (amongst others, Texas Instruments MSP430 & MSP432, Atmel AT91SAM9G20, Atmel ATmega328P, ARM Cortex-M4 MK20DX256, MicroChip PIC32MM0064FPL028). This should however be checked by the developer before choosing a micro-controller. The baud rate is set at 1 MHz for CS14 and 500 kHz for PQ9. As provided in Section 3.4.2, this yields approximately 600 kbit/s of effective (message) data throughput for CS14. The clock at any node connected to the bus must be within $\pm 0.5\%$ of these values to ensure reliable communications. It is recommended to select the general clock frequency of the micro-controller at least 16 times higher than the baud rate (this yields 8 MHz for PQ9 and 16 MHz for CS14).

Figure 3.21: Proposed byte definition for RS-485.

The transaction flow is determined by the master sending a packet to the slaves, after which the slave responds with an acknowledgement or return packet. The packet is defined in Figure 3.22. It starts with a *destination address* to define which node should receive a packet. It also specified the *source address*, which is used as check of the proper transaction flow. The address of the master is [0x00]. A general call to all slave nodes can be made using [0xFF] as address, which is a broadcast without response by any of the slaves. There are 253 potential slave nodes, which is deemed more than sufficient for PocketQubes and CubeSats. The *message size* byte defines the length of the message, with a maximum of 255 bytes. Slaves return a message size of [0x00] as acknowledgement of commands which do not require data return. A 16-bit CCITT *CRC* is added to provide robustness. It

is initialized on the seed [0xFFFF] and calculated on the complete packet prior to the CRC, using least significant bit first [23].

Figure 3.22: Proposed packet definition for RS-485.

When the command is not acknowledged, data is not returned or the CRC of the return message is incorrect, the transaction is considered to be faulty. The maximum time-out is 100 ms, including latency and time intervals between bytes. The counter starts at the master after the destination address byte is sent and by the slave when it identifies its own address (or the general address). When the entire transaction is not completed within this time, both slave and master discard the information which is sent. The master will internally flag that the transaction is timed-out, such that it could potentially be used by the application layer for error handling. Given that all transactions are initiated by the master, error handling after failed transactions does not have to be standardized. It is recommended to at least include one retry by the master.

3.8. Conclusions

A proposal for an electrical interface standard for CubeSats and PocketQubes has been established. The main objective was a lean standard which meets expected future demands as opposed to existing versatile standards which exhibit the risk of incompatibility between subsystems from different developers.

Based on the defined set of selection criteria, community survey input and the AHP trade-off method, RS-485 is favored as housekeeping data bus for both PocketQubes and CubeSats. Tests results show that it outperforms I^2C, dI^2C and CAN in terms of power and effective data throughput. In terms of robustness features, it comprises differential signaling, but a low level of hardware control. In terms of legacy support is scores relatively low, but this is a criterion which can easily be improved in the future if the proposed electrical interface is adopted by multiple parties. For a future study it is recommended to test the RS-485 bus for very high data rates such that it can be used as a point-to-point payload data bus for demanding payloads (or RS-422, which is very similar in point-to-point configuration), data storage and high speed radio transmitters. Also, it is recommended to perform in-orbit tests with self-powered sensors over a wireless Bluetooth Low Energy connection to be able to reduce wiring harness to components which cannot be integrated in the internal stack of subsystems.

Power distribution can best be done by supplying the unregulated battery voltage over switched and protected lines to (groups) of subsystems. This limits the number of pins used and reduces conversion losses. Power protection features and duty cycling of subsystems to save power can be implemented at the central

EPS unit. Together with the chosen data bus, this yields a 9-pin (PQ9) and 14-pin (CS14) standard electrical interface for PocketQubes and CubeSats respectively. PQ9 has only 15% of the electrical interface lines compared to PQ60, CS14 only 13% compared to PC/104. This saves in both cases significant board space, but more importantly leads to a very lean interface which will lower risk for incompatibilities between physical subsystems. However, it comes at the cost of freedom for developers to choose a data bus and power distribution architecture to their specific needs and preferences.

An important assumption made in this study is that CubeSats and PocketQubes do not use redundancy for main subsystems. While the proposed interfaces do not prohibit this per se, the lack of a redundant data bus and the limited amount of power distribution lines make a true single-point-of-failure free design impossible. A follow-up study is recommended to investigate the impact on the overall reliability for these small satellites under these assumptions.

A topic not addressed in this paper is the development of (mega) constellations of very small satellites. Present day examples are the Flock CubeSats from Planet [24] and the Lemur CubeSats from Spire [25]. In relation to an electro-mechanical interface standard, it is expected that technical criteria are more important than community support. The rationale behind this expectation is that the main players have sufficient finances to develop many iterations of the spacecraft before the full operational constellation is deployed and have the financial means to optimize the satellite and when necessary customize subsystems and even the interfaces to enhance the performance of the satellite. Next to this, the financial aspect of series production in relation to the electro-mechanical interface becomes important. The proposed PQ9 and CS14 interface standards can be implemented with just a few very cheap components (few Euros/Dollars) and assembly will take only a few minutes by a solder expert or can even be fully robotized.

As a follow up of this study, the electrical characteristics of the reset line are defined and testing has been performed with engineering models of PocketQube systems using the PQ9 interface. This has led to the public release on the new interface standards PQ9 and CS14 [22].

References

[1] J. Bouwmeester, S. P. van der Linden, A. Povalac, and E. K. Gill, *Towards an innovative electrical interface standard for PocketQubes and CubeSats,* Advances in Space Research **62**, 3423 (2018).

[2] S. Busch, *CubeSat Subsystem Interface Definition (v0.3)*, Tech. Rep. (UNISEC Europe, Wuerzbrug, 2015).

[3] N. Cornejo, J. Bouwmeester, and G. Gaydadjiev, *Implementation of a Reliable Data Bus for the Delfi Nanosatellite Programme,* in *Proceedings of the 7th IAA Symposium on Small Satellites for Earth Observation* (IAA, Berlin, 2009).

[4] V. Riot, L. Simms, D. Carter, T. Decker, J. Newman, L. Magallanes, J. Horning, D. Rigmaiden, M. Hubbell, and D. Williamson, *Government-owned Cube-*

Sat Next Generation Bus Reference Architecture, in *Proceedings of the 28th AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2014).

[5] E. Becnel, S. McAndrew, L. Strass, T. Walkinshaw, and K. Worrall, *PQ-60 Standard Document (v1.1)*, Tech. Rep. (2015).

[6] D. Selva and D. Krejci, *A Survey and Assessment of the Capabilities of Cubesats for Earth Observation,* Acta Astronautica **74**, 50 (2012).

[7] A. Poghosyan and A. Golkar, *CubeSat Evolution: Analyzing CubeSat Capabilities for Conducting Science Missions,* Progress in Aerospace Sciences **88**, 59 (2017).

[8] C. W. de Boom, N. van der Heiden, J. Sandhu, H. C. Hakkesteegt, J. L. Leijtens, L. Nicollet, J. Bouwmeester, G. van Craen, S. Santandrea, F. Hannoteau, C. W. D. Boom, N. V. D. Heiden, J. Sandhu, H. C. Hakkesteegt, J. L. Leijtens, L. Nicollet, J. Bouwmeester, G. V. Craen, S. Santandrea, and F. Hannoteau, *In-Orbit Experience of TNO Sun Sensors,* in *Proceedings of the 8th International Conference on Guidance, Navigation and Control* (ESA, Karlovy Vary, 2011).

[9] I. Arruego, J. Rivas, J. Martinez, A. Martin-Ortega, V. Apestigue, J. R. De Mingo, J. J. Jimenez, F. J. Alvarez, M. Gonzalez-Guerrero, and J. A. Dominguez, *Practical Application of the Optical Wireless Communication Technology (OWLS) in Extreme Environments,* in *Proceedings of the IEEE International Conference on Wireless for Space and Extreme Environments*, edited by IEEE (Orlando, 2015).

[10] R. Amini, G. Gaydadjiev, and E. Gill, *Smart Power Management for an Onboard Wireless Sensors and Actuators Network,* in *Proceedings of the AIAA Space Conference and Exposition*, edited by AIAA (Pasadena, 2009).

[11] R. Schoemaker and J. Bouwmeester, *Evaluation of Bluetooth Low Energy Wireless Internal Data Communication for Nanosatellites,* in *Proceedings of the 4S symposium* (ESA, Valetta, 2014).

[12] ITU, *ITU-T Recommendations X.200*, Tech. Rep. (ITU, 1994).

[13] F. Leens, *An Introduction to I2C and SPI Protocols,* IEEE Instrumentation and Measurement Magazine **12**, 8 (2009).

[14] NXP Semiconductors, *PCA9615 (Datasheet)*, Tech. Rep. 1.1 (NXP, 2016).

[15] W. Lawrenz, *CAN System Engineering*, edited by W. Lawrenz (Springer-Verlag, London, 2013).

[16] M. Soltero, J. Zhang, C. Cockril, H. P. a. Industrial, K. Zhang, C. Kinnaird, and T. Kugelstadt, *RS-422 and RS-485 Standards Overview and System Configurations*, Tech. Rep. (Texas Instruments, 2010).

[17] T. L. Saaty, *Decision Making With the Analytic Hierarchy Process,* International Journal of Services Sciences **1**, 83 (2008).

[18] J. Bouwmeester, R. Amini, and R. Hamann, *Command and Data Handling Subsystem for a Satellite Without Energy Storage: Delfi-C3,* in *Proceedings of the 58th International Astronautical Congress* (IAF, Hyderabad, 2007).

[19] J. Bouwmeester and J. Guo, *Survey of Worldwide Pico- and Nanosatellite Missions, Distributions and Subsystem Technology,* Acta Astronautica **67**, 854 (2010).

[20] J. Bouwmeester, *Dataset: Survey Results for CubeSat and PocketQube Data Bus Trade-off,* (2017).

[21] K. D. Goepel, *Implementing the Analytic Hierarchy Process as a Standard Method for Multi-Criteria Decision Making In Corporate Enterprises – A New AHP Excel Template with Multiple Inputs,* in *Proceedings of the International Symposium on the Analytic Hierarchy Process* (Kuala Lumpur, 2013).

[22] J. Bouwmeester, *PQ9 and CS14 Electrical and Mechanical Subsystem Interface Standard for PocketQubes and CubeSats*, Tech. Rep. (Delft University of Technology, 2018).

[23] N. Ross, *A Painless Guide to CRC Error Detection Algorithms (v.3),* (1993).

[24] C. R. Boshuizen, J. Mason, P. Klupar, and S. Spanhake, *Results from the Planet Labs Flock Constellation,* in *Proceedings of the 28th AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2014).

[25] E. Hand, *CubeSat Networks Hasten Shift to Commercial Weather Data,* Science **357**, 118 (2017).

4

Innovative Concepts of Physical Architectures

No one wants to learn by mistakes,
but we cannot learn enough from successes
to go beyond the state of the art.

Henry Petroski

The dominant architectural approach in CubeSats and PocketQubes is the use of modular physical units, each hosting components of classical subsystems on one or more dedicated printed circuit boards. Many of these small satellites, however, also host subsystems or experiments with a slightly alternative approach, e.g. with cellularization of components or the integration of functions from different virtual subsystems into a single physical unit. These concepts have been also investigated and proposed by some studies on a rigorous implementation. Examples of this are a fully celullarized satellite, a satellite comprising only outer panels and a satellite using only plug-and-play technology. While they offer promising advantages when implemented smartly as part of a new architecture, their disadvantages become dominant when such a concept is implemented in a too rigorous and dogmatic manner. A smartly chosen hybrid of several concepts is investigated. Panels on the sides of the satellite body integrate all components which need to be exposed to the environment, including related electronics, in a cellular fashion. Internally, modular systems are still used, but some classical core subsystems can be integrated towards a single core unit. A lean approach on redundancy and electrical interfaces saves volume (for more payload volume or smaller satellites) and reduces overall systems complexity.

Parts of this chapter have been published in Journal of British Interplanetary Society **71**, 7 (2018) [1]

4.1. Introduction

The physical architecture of a satellite is the foundation on which all its functions are built upon. It determines the breakdown of a satellite in physical subsystems and components, the physical location of these units and the structural and electrical interfaces between them. In terms of technology, the large use of commercial-off-the-shelf electronics in CubeSats and PocketQubes is one of the major differences compared to larger satellites. These satellites are developed in a modular fashion using standard interfaces and a physical breakdown along the breakdown of traditional (virtual) subsystems also used in larger satellites. In Section 4.2, the physical architecture of a few CubeSats and PocketQubes is analysed to provide an overview of common practices and noteworthy differences of state-of-the-art CubeSats. In Section 4.3, a further step in the analysis is made by providing an overview and reflection on potential future advanced architectural concepts. In Section 4.4, several of these concepts are worked out with examples for practical insight. In Section 4.5, a study case is presented using a subset of advanced ideas to show the impact on design, complexity and payload volume. Finally, conclusions are provided in Section 4.6.

4.2. Survey of State-of-the-Art Architectures

In this section, examples from literature on the architecture of CubeSats and PocketQubes are provided. The aim is to identify the common practices as well as highlighting a few remarkable aspects related to their physical architecture.

ArduSat-1 and ArduSat-X are open-source single unit (1U) CubeSats comprising an optical spectrometer and camera and several other sensors [2]. They were the first satellites launched by the company Spire (formerly known as NanoSatisfy). The physical architecture uses a stacked approach with PC/104 compatible units for the On-Board Computer (OBC), Electrical Power Subsystem (EPS), a radio transceiver and an antenna board. The most remarkable item is a payload processor module which holds an ATmega2561 supervisor processor and 16x ATmega328 processor nodes on a single board, all of them Arduino compatible. Arduino is an open source electronic prototyping platform using a standard set of micro-controllers and has a wide community support. This approach allows for distributing experiments to student teams. The relative payload volume is about half of the satellite according to Figure 2 in reference [2].

BeEagleSat is a 2U CubeSat developed by the Istanbul Technical University in the framework of the QB50 project [3]. Its payloads are the QB50 'multi needle Langmuir probe and thermistors' suite and an X-ray detector. It comprises several physical subsystems from different manufacturers for power, attitude control and high speed radio communication. The main interface is based on the PC/104 connector. The most remarkable subsystem is the 'OBCOMS' which is a single board comprising both an on-board computer and a beacon radio. The relative payload volume is about one-third of the satellite, according to Figure 1 in reference [3].

ESTCube-2 is a 3U CubeSat for the demonstration of Coulomb drag propulsion, a multispectral imager and advanced communication payloads [4]. Noteworthy in

its physical architecture is that the outer structural panels of the satellite comprise both solar cells as well as the maximum power point tracking circuitry and a sun sensor by using aluminum printed circuit boards as substrate. Also, there is tight integration of core bus subsystems where several virtual subsystems are sharing a few onboard microcontrollers. This integrated bus uses 0.5U of volume.

Galassia is a 2U CubeSat with a Total Electron Count payload and a quantum entangling demonstration payload. It has a standard modular physical architecture, comprising PC/104 based Printed Circuit Boards (PCBs) for OBC, EPS, passive attitude control, radio transceiver and the payloads [5]. The relatively simple bus subsystems requires about 1U, half of the satellite, in total.

GOMX-4 from GomSpace is a standard satellite platform for 6U CubeSats [6]. Its physical architecture is exemplary for the modular approach in which many CubeSats are developed. This approach means that each virtual subsystem typically has one or more physically distinct units which are connected through a standard electrical interface (in this case a PC/104 connector). The most remarkable part of this architecture is the Software Defined Radio (SDR) which is used for the Inter Satellite Link (ISL), high speed transmission to ground and the reception of Automatic Dependent Surveillance-Broadcast (ADS-B) signals from airplanes. This architecture offers an integrated platform which could be used both for advanced bus and payload purposes. The fact that a large part of the functionality resides in software, means that a standard unit can be (re-)configured and aggregated for different communication functions.

Delfi-C^3 [7] and Delfi-n3Xt [8] CubeSats (both 3U) from Delft University of Technology (TU Delft) have been launched in 2008 and 2013 respectively. In terms of architecture, both follow a modular subsystem approach similar to GOMX-4. However, both satellites attempted to provide a no single-point-of-failure design. On Delfi-C^3, a backup mode was created with analogue measurements of the thin film solar cell technology demonstration payload. In lack of time, priority was given to the nominal mode and the backup mode was not properly tested and the ground segment not completed. In its almost ten years of operation, the backup mode was never needed to continue critical operation but was activated a few times, most likely due to a false trigger.

Delfi-n3Xt (shown in Figure 1.3) used a more classical redundancy concept, in which critical systems were duplicated. The internal stack is shown in Figure 4.1. However, on the data bus interface single-point-of-failures could not be completely mitigated and after three months of operations, having completed the primary mission objective, the satellite became silent after attempting to switch on a radio transponder as discussed in Section 2.3.1. The architecture of Delfi-n3Xt is further elaborated in Section 4.5.

To date, few PocketQubes have been launched, so information on their architectures is scarce. A website on the 1p WREN PocketQube [9] reveals that the outer structure, typically an aluminum plated box on CubeSats, has been completely removed. The small size of the satellite makes it possible that launch loads are completely handled by internal rods and/or by PCBs used as outer panels. WREN and the UoMBSat1 PocketQube of the University of Malta [10] both show that still a

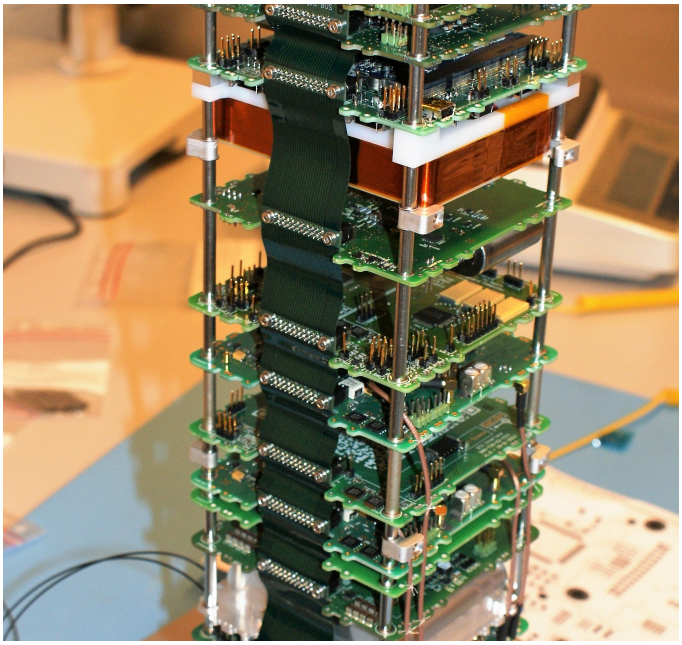


Figure 4.1: Delfi-n3Xt internal stack.

modular stack of PCBs is used to host the subsystems.

Beside the scientific references, a survey of websites, pictures and hardware displayed on conferences reveals that a vast majority of CubeSats and PocketQubes are internally built on a modular stack of PCBs. Typically, each of the functional subsystems is represented by one or more physical PCBs. While payload volume differs significantly between the satellites, a stack of PCBs takes significant volume and the height of the connector and number of subsystems drives the total volume consumption of the spacecraft bus. The dominant architectural approach of mapping functional (virtual) subsystems (such as the electrical power subsystem, the command and data handling subsystems, etcetera) to one or more distinct physical units which are placed in an internal stack, may be challenged by some innovative concepts.

4.3. Survey of innovative architectural concepts

Next to the survey on CubeSat and PocketQube missions, a literature study of several reference papers has been carried out which addresses innovative architectural concepts specifically. A summary of the literature is provided in this section, followed by a qualitative analysis on its main advantages and disadvantages.

4.3.1. Cellular Concept

Cellularized satellites have been proposed to "achieve cost savings, flexibility and reliability while maintaining the overall mission performance" by the introduction of "satlets" [11]. A distinction is made between single-function satlets and system satlets. The single-function satlets comprises standard modular pieces which can be combined to meet the mission specific requirements. A given example is the use of spatially distributed reaction wheel assemblies, which together provide the total torque and momentum storage. System satlets can be regarded as a module which integrates several subsystem functions such that it can operate as an independent system. An example of a physical breakdown is shown in Figure 4.2, which comprises a modular connectable nano-satellite-scale package which integrates core satellite functions such as electrical power acquisition and storage, attitude determination and control and computational processing.

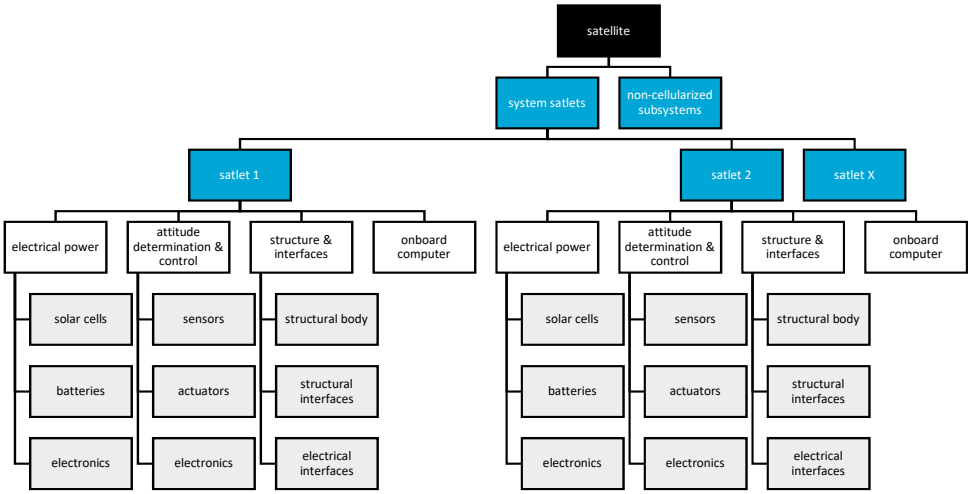


Figure 4.2: Example physical breakdown of a cellularized satellite using 'system satlets'.

The resources can be shared with the rest of the satellite in a building-block fashion. The benefits mentioned are thought to be acquired with the aid of mass production and integration in many satellites of these standard building blocks. A demonstration of this concept was launched by the end of 2018 on the eXCITe mission which comprises 14 of the HISat blocks together with several payloads, deploy-able solar array and high data rate communication radios.

The satlet concept is relatively simple to comprehend and implement. Its advantages are the ability to scale up the technical capacity of the satellite with mission demands and the potential to increase reliability by introducing the option for graceful degradation. Its main disadvantage is that system efficiencies (in terms of power, mass and volume) are lower compared to larger subsystems or components. The single-function cellular concept will be investigated further in Section 4.4. The system level satlets combines integration of several satellite core functionalities of subsystems with cellularization. An additional disadvantage is that this

concept severely restricts physical configuration options and fixes the ratio of the technical specifications. For example, if a mission requires the equivalent computational power of ten satlets, the satellite would also receive ten times the satlet data storage, ten similar attitude sensors and actuators, ten times the solar cells, even if this is not truly needed. Also one can question the added benefit of a satlet with solar cells, if one still adds a non cellular deployable solar array like in the eXCITe mission example. However, aspects of the system satlet concept may still be attractive to investigate, such as the integration of satellite core functions into a single physical unit. CubeSats and PocketQubes always have six sides of the body. This fact can be used to investigate system satlets which integrates components and satellite functions which are typically residing on each side, such as sun sensors. But also potentially omni-directional radio communication could be worth to investigate. Finally, an attractive option could be to use PocketQube sized components and systems as cells for CubeSats.

In another study [12] it was found that a physical architecture based on an OBC with a single-master data communication bus exhibits a relative high number of failures (40% of these CubeSats were never heard on ground), followed by an OBC connecting via separate buses to subsystems. The best statistics were provided by CubeSats based on a distributed design using a multi-master bus, for which 80% of the CubeSats fulfilled (part of) its mission objectives. The same study also investigated correlation between mission success and the number of redundant subsystems (up to three) which are regarded as critical (i.e. OBC, EPS, communications). Only a weak correlation is found, since with two redundant subsystems the reliability seems to increase with respect to a singular system, but a slight decrease is seen with three w.r.t. two redundant subsystems. This correlation is used as a key argument to propose a cellular architectural concept. The concept presented by Erlank & Bridges [12] is however different from the satlet concept: they propose the use of Artificial Stem Cells (ASCs), based on the analogy of biological cells.The ASC comprises non-volatile memory (DNA), a central micro-controller (macro-molecular machinery) and several micro-controllers with generic input and outputs (proteins) to perform tasks and connect to the outside world.

The practical application is demonstrated on SME-SAT by a four protein cell (Figure 4.3), each of the proteins used to drive an identical Control Moment Gyro (CMG) and a different small technology demonstration payload. This is just a very simple demonstration, since the intended architecture would consist of multiple cells, with proteins of different cells being cross-strapped with devices (such as gyros) using multiple different communication busses.

The study on the ASC concept advocates and clearly explains the use of cellularization for graceful degeneration. However, it also states that reconfiguration of the ASC function, the communication paths and potential cross-strapping payloads between the ASCs has been considered but not implemented as it "was deemed unnecessarily complicated" for the SME-SAT mission. The paper fails to describe how higher level satellite functions could be implemented as ASCs in a reliable and practical manner, which gives rise to the question whether the biological analogy can really be followed. The complexity of DNA and cells in biology is tremendous

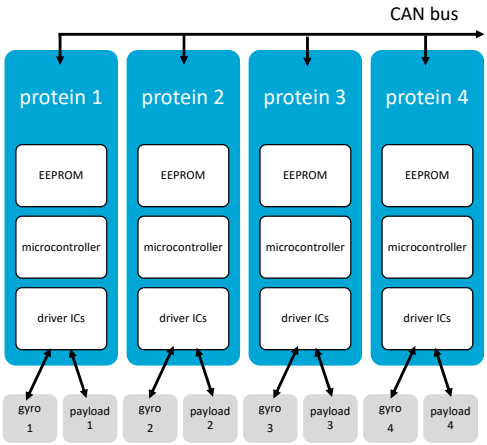


Figure 4.3: Sketch of a four protein ASC configuration.

and not yet fully understood. Also, in biology there is a physical mobility of cells which is very difficult to mimic with its technical counterpart. The benefits of mixing attitude control actuators and payloads to a single ASC in the example seems arbitrary and is not explained. Erlank & Bridges [12] continue with a bench-top demonstration of a complete ASC based attitude control subsystem. The presented design is too complex to be summarized in a simple way here, nevertheless the main conclusion drawn in the paper is that a reliability increase of the system can be expected mainly due to potential reconfiguration of the software tasks of proteins and the graceful degeneration features of the concept. It however comes at the expense of significantly higher power consumption (+77%) and higher complexity compared to a traditional design. While the concept of ASCs is theoretically interesting, it is too far fetched to implement in the near to mid-term future and it is not yet clear if the benefits on the long term will outweigh its costs.

4.3.2. Panel Concept

A 'nano-modular format' (NMF) has been proposed for CubeSats which focuses on a different structural integration concept [13]. The six faces of a CubeSat form the basis which comprises a structural outer panel with hinges towards the other faces and holds part of all internal equipment which can be placed in a pyramid-shaped envelope. A 1U CubeSat thus always consists of six physical distinct units, while for the larger CubeSats the configuration can be extended by using 1U units placed side-by-side or by using a larger base panel. The hinges and electrical connections between the panels are supposed to quickly integrate panels towards a complete satellite. An artist impression of the envelope of a 1U NMF panel is provided in Figure 4.4.

The concept is limiting the amount of distinct physical units to a fixed number or range (6 for 1U, 6-10 for 2U), while each unit takes a fixed envelope of space. The pyramid shaped envelop is considered to be impractical, for example for housing a

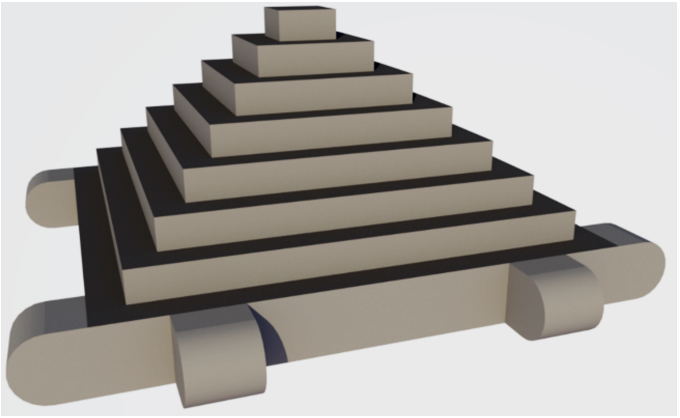


Figure 4.4: Artist impression of 1U NMF envelope.

propellant tank. An interesting part of this concept is however the ability to quickly integrate the satellite with a limited number of steps. The severe reduction of manual integration steps for wiring externally located components (solar cells, sun sensors, antennae, etcetera) to internal units, as compared to a standard modular stack approach, is an idea which can be considered for a new architectural concept.

4.3.3. Plug-and-Play Concept

The same authors who discuss the panel concept also introduces the concept of Space Plug-and-Play Avionics (SPA) for CubeSats [13]. SPA is a data driven architecture, in which modular equipment can be added to the satellite with simple electrical and software interfacing, thanks to the use of standard command and data handling approaches and embedded electronic data-sheets. It can best be understood by looking at the way how (peripheral) equipment of computers with an USB interface can be used almost directly after connection without the need of manual installation of software drivers. SPA is implemented on several CubeSats and mentioned by many references, which are amongst others the Trailblazer [14] and TechEdsat [15] CubeSats. The electrical interfaces of SPA come in incremental steps. The SPA-1 interface is specifically designed around the I^2C data standard and comprises a four-pin wiring harness with just I^2C and 5V power. It is a minimalist SPA interface aimed at CubeSats [16]. Higher performance SPA interfaces are SPA-U (based on USB), SPA-S (based on SpaceWire) and SPA-O (optical). The general SPA physical architecture relies on central hub or routers to connect all equipment and local Remote Terminal Units (RTU) or Appliqué Sensor Interface Module (ASIM) to interface and describe the software specification and behavior.

When consulting literature about the implementations of SPA, various different terms are used and the concept seems to have evolved over time and branched off into a Swedish and US based version. This leads to confusion, for example when the terms RTU and ASIM are used for a seemingly same functional unit. The key philosophy behind the software architecture fills a gap in terms of interface

standardization. The lean electrical interface for components is also considered to be an advantage. However, many other aspects are considered to complicate the development of subsystems and components even if the final integration would be fluent. The use of RTUs/ASIMs may simplify the development, but may also add volume and power consuming electronics for the very small satellite components typically found in PocketQubes and CubeSats. A reflection of 10 years of Plug-and-Play (PnP) development provides insights in the evolution, successes and critics of the standard [17]. It states: "To the critics of SPA, however ASIMs were viewed as adding complexity and overhead, when in fact the intent was the opposite." This means that there is an acceptation problem of PnP outside its developers community on aspects of the standard. Also it becomes clear from the reference [17] that the standard has not yet fully matured and that many goals of PnP have not yet been achieved. What can be learned from SPA concepts is that it would be valuable to specify one or a few lean electrical interface standards for PocketQubes and CubeSats. Separately, a command and data handling standard can be developed in line with the PnP philosophy, in which the housekeeping data, the commanding and the specification of components is completely and uniformly described in a hardware abstraction and service layer code, such that it can be handled by application layer software in an autonomous and transparent manner. The parallel development of a public electrical interface standard and an open source PnP standard software will facilitate the maturation of the standards on their own pace and provides a higher chance for acceptation than a single combined solution which requires a too disruptive transition and a vendor lock-in.

4.3.4. Lean Electrical Interfaces

Electrical interfaces are a dominant aspect of modularization and can have a significant impact in the available volume. In Chapter 3, subsystem interfaces are already extensively investigated and a new lean interface is proposed. One step further from wired subsystem and component interfaces are devices which are self-powered and have a wireless interface. The lack of wiring harness saves volume and potentially also reduces integration complexity.

On the Delfi-C^3 satellite, a sun sensor from TNO is demonstrated which acquires its power with a local solar cell and transmits its data over a wireless radio link [18] (shown in Figure 4.5). In a recent study, a proof-of-concept temperature sensor is developed which can power itself by using a thermal electric cell with only 2.3 K of temperature difference between both sides of the sensor [19]. Communication of this sensor is via a Bluetooth data link. This type of self-powering sensors exhibits even larger freedom in placement. Magnetometers would also be an interesting type of sensor as they could be placed away from power electronics or a few can be spread over the satellite to be able to filter out locally generated noise.

The advantages of autonomous wireless devices increase with satellite size as wiring harness increases. On PocketQubes and CubeSats they have less impact on this aspect. Also, the volume available on a large satellite would enable a larger power acquisition unit which can be used for more demanding sensors and actuators. Disadvantages of self-powered wireless sensors are the following: they may

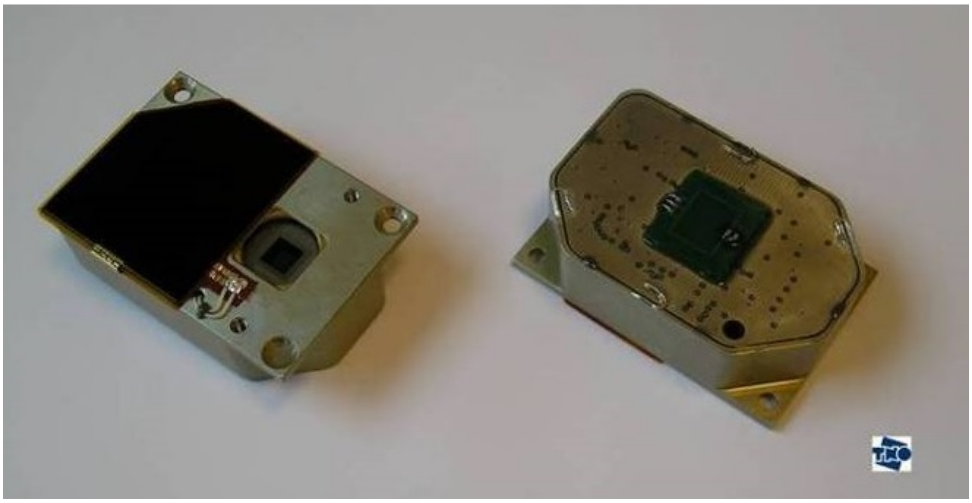


Figure 4.5: Delfi-C^3 Autonomous Wireless Sun Sensor (TNO).

cause radio interference to other radio based systems or to each other, they are dependent on a conditional power source (sunlight or thermal gradient) and they are larger and more complex than integrated sensors on-board existing subsystems or panels. Within the scope of this thesis, focusing on very small satellites, only low power sensors with specific placement requirements for which the integration of the wiring harness is relatively complex would be good candidates to consider for very small satellites.

4.4. Concept Analysis

In this section, some of the concepts are investigated with the aid of a few examples.

4.4.1. Cellular Reaction Wheels

At TU Delft, a reaction wheel has been designed for the 3U CubeSat Delfi-n3Xt [20] and for the 3p PocketQube Delfi-PQ [21] as can be seen in Figure 4.6. Both are highly optimized designs in terms of volume and power consumption, while they provide torque and momentum storage required for their respective size in Low Earth Orbit.

To match the momentum storage of a single CubeSat reaction wheel, in total 15 PocketQube reaction wheels are needed for a cellular configuration. The comparison is provided in Table 4.1. The total volume is about five times higher for the cellular approach, mainly due to nonlinear scaling of the housing and motor. An orthogonal set of cellular reaction wheels (so 45 in total) would consume a minimum volume of 17% of a single unit CubeSat, not including inter-spacing and mounting losses. This does not render the concept infeasible. The full range torque of the cellular approach is slightly lower than for the single CubeSat reaction wheel. How-

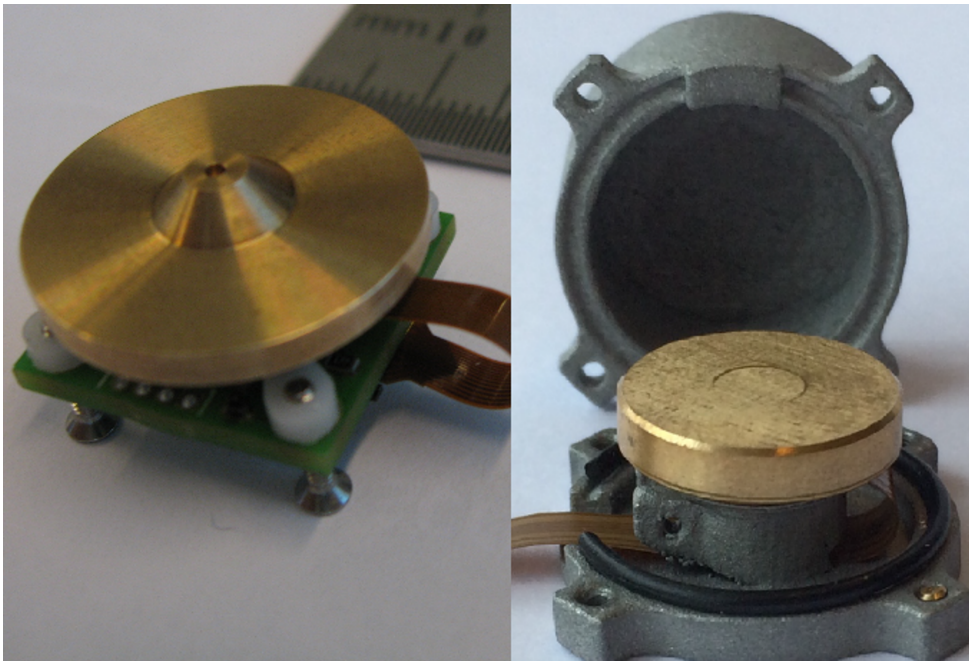


Figure 4.6: Delfi-n3Xt (left) and Delfi-PQ (right) reaction wheels.

ever, this only applies in the region near the maximum momentum storage, which for the cellular approach means that all reaction wheels are almost saturated. The chance that a maximum torque is needed in that region is fairly small and can be neglected. Regarding the power consumption, it seems that the minimum power (the power at a low nominal rotation speed) is better for the cellular approach, while the single reaction wheel is better at the maximum momentum storage. However, in a cellular approach it would be possible not to turn on all the reaction wheels at a time, which may yield a significant lower average power consumption. Also the disruptive torque at zero speed crossing (due to static friction), may be compensated in the cellular approach with a proper combined acceleration of a few other reaction wheels. Finally, the cellular approach provides a more fine torque control. Overall, it can be concluded that the cellular approach is costly in terms of volume and also potentially in terms of finance. On other technical aspects it is however an interesting concept which introduces opportunities for increased reliability by graceful degradation, more accurate control and average power optimization.

4.4.2. Cellular Magnetorquers

There are two types of magnetorquers which are typically found in small satellites: those with a permeable core and those without. A permeable core strengthens the creation of a dipole moment by aligning the magnetic field lines. The 'air-coils' have no such medium. The magnetic dipole moment m relates to the number of

Table 4.1: Specification of Reaction Wheels.

	1 CS RW	1 PQ RW	15 PQ RW
torque (full range)	$5.5 \cdot 10^{-6}$ Nm	$3 \cdot 10^{-7}$ Nm	$4.5 \cdot 10^{-6}$ Nm
momentum storage (one-way)	$1.6 \cdot 10^{-3}$ Nms	$1.1 \cdot 10^{-4}$ Nms	$1.6 \cdot 10^{-3}$ Nms
volume	11 cm^3	4 cm^3	58 cm^3
power (min.)	177 mW	4 mW	60 mW
power (max.)	237 mW	25 mW	375 mW

windings n, the electrical current I, the enclosed area A and the core gain factor k as in the following simplified equation:

$$m = k \cdot n \cdot I \cdot A. \tag{4.1}$$

The gain factor k for a core-less magnetorquers is set to 1 and for a permeable core it is, within the boundaries of a small satellite, positively related to the length of the core. With core-less magnetorquers, typically the enclosed volume is maximized to make it most efficient in terms of volume (of the copper wiring) and power. For magnetorquers with a core, typically the length of the rod in increased to make it more volume and power efficient.

In case of a cellular approach, there would be no difference in volume and power efficiency when the core-less magnetorquers would be of equal enclosed area or if the core rods would be aligned. The advantage here would be the option of graceful degradation if one of the drive electronics would fail. The disadvantage is that more drive electronics is needed which increases the volume and complexity on a higher system level.

If more freedom is desired in configuration, smaller and non-aligned magnetorquers are required. For a cellularized square core-less magnetorquer towards four cells of half the diameter of the original, using the same amount and thickness of wiring, the total power consumption for a given dipole moment will double. For a torque rod, cellularization by simply 'cutting' it in smaller pieces along the rod axis will also negatively impact the total power consumption.

4.4.3. Solar Power Acquisition Units

In many CubeSats, solar cells are mounted on a panel and connected to an internal Electrical Power Subsystem (EPS) unit which hosts Maximum Power Point Trackers (MPPT) or circuitry using other power conversion methods. The MPPT circuits on the EPS unit are limiting the amount and/or combination of solar arrays which can be connected. An alternative idea is to integrate the solar cells on a PCB and host the MPPT circuitry on the backside of this PCB. With protective diodes, these 'solar power acquisition units' can be connected to a main distribution bus in a safe manner. Next to this, the unit can host a monitoring circuit to determine the local voltage, current and temperature. This would require an additional connection to a (linear) data bus to the internal OBC or EPS. This concept is similar to the circuit on a typical EPS unit, but the main difference is the physical location. It allows a

cellular approach in which the total solar array can be scaled up and assembled out of standard units according to the mission needs and the preferred configuration. Potential advantages are: the use of standardized (mass produced) units, the option for graceful degradation, less susceptibility to local shadowing and less limitations on the potential combinations and configurations of solar panels. The (potential) disadvantages are: an increase in the total amount of circuitry, the need for a data bus connection to the outer panel and the need for holes in the outer structure (if present) at the location of the circuitry.

For Delfi-PQ, units with two 80 mm x 40 mm triple-junction solar cells of 30% efficiency are currently being developed which can be compared to a theoretical eight-cell panel for a CubeSat connected to a standard EPS unit. The ST SPV1040 integrated circuit is chosen as MPPT and provides a single cell Li-ion battery output voltage, with an efficiency between 93% (at 2.5 W input power) and 97% (at 0.25W input power) when using two cells in parallel. In fact, one can even use this device for a single solar cell with 94% at 1.2 W input power. These efficiency ranges are very similar to those of a CubeSat EPS unit with MPPTs on an internal stack board. For instance the GOMSpace NanoPower P31 has a power efficiency between 93% (at 9.5 W input power) and 96% (at 1 W input power) [22]. Replacing a body mounted CubeSat solar panel with four solar power acquisition units is thus possible without a penalty in power efficiency.

4.4.4. Cellular Flat Radios

For Delfi-n3Xt, a 2.4 GHz radio was developed which contained the patch antenna and the electronic circuit on the same PCB [8]. This directional radio transmitter system (STX) was supposed to be used for relatively high data rate transmission (up to 1 Mbit/s). It has a total height of about 5 mm except for the connector. It was mounted on top of the structural outer panel and did not consume useful volume within the satellite. However, an interface board (of 14 mm CubeSat stack height) in the internal stack was required to connect the standard interface of the internal stack to the STX. Delfi-n3Xt also has redundant radio transceivers acting on a down-link at 145 MHz and an up-link at 435 MHz. The CubeSat stack height of each PCB is 20 mm. These are connected to a shared antenna system comprising of four deployable antennae of about 0.5 m in canted turnstile configuration with a near omni-directional radiation pattern. This antenna system and the deployment board consume 41 mm of total stack height. The purpose of this redundant radio transceiver system is to provide reliably transmission and reception of telemetry and tele-commands under all circumstances, including a tumbling satellite. This redundant system consumes about 0.8U of a CubeSat and the total communication subsystem almost 1.0U when the STX interface board is included. It would therefore be interesting to find a concept which integrates the advantage of a directional patch antenna with back-side electronics with the ability to provide near omni-directional communication for the tumbling and safe modes of the satellite. One possibility is to have a directional flat transceiver on each side of the satellite, similar to the STX, but with a higher degree of software configurability. In safe mode, all radios will transmit the same telemetry message simultaneously (e.g. in “beacon mode”)

either in a side-by-side band operation or in a spread spectrum configuration. With six orthogonal patch radios, the minimum gain would be achieved at 55° from its normal. The radiation pattern of the STX, provided in Figure 4.7, yields a minimum gain of +2 dB at 55°. Since the electrical input power is divided over six radios, the radio frequency output will be 8 dB less (assuming that almost all electrical power goes towards the radio amplifier and its efficiency is fixed) than its singular counterpart. Compared to a singular perfectly omni-directional (isotropic) transceiver, this would yield -6 dB as worst case output. This is comparable with the worst case output of a canted turnstile configuration on the 435 MHz band on Delfi-n3Xt which was designed for omni-directionality.

When ground station pointing is achieved, the communication will switch to a single patch for transmission which can occupy a wider bandwidth and/or increased transmission power at a higher data rate. In the STX example, this would yield a gain of +9 dB.

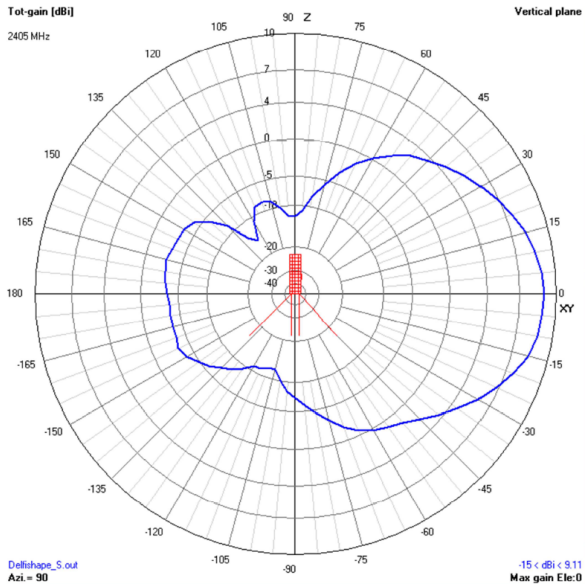


Figure 4.7: Radiation pattern STX.

For this concept, a high degree of software configurability is required including change of frequency, modulation and data rate. Also the transmission power should be able to change with equal power added efficiency. Furthermore, for the omni-directional mode, a very good channel separation is essential to avoid that they mutually increase each others noise floor. If the interface towards the rest of the satellite could be lean (so no complete interface board required), the whole communication system in this concept would not consume considerable internal volume, would not require complex deployment systems and would potentially increase reliability by providing the option for graceful degradation. The concept could be further developed with phased array antennas, for which the potential directional

gain can be increased and even be made independent of attitude orientation.

4.4.5. Advanced Integrated Outer Panel

While the solar power acquisition units presented in Section 4.4.3 are a relatively small step from the traditional approach, the concept can act as baseline for a more advanced outer panel approach. Solar cells, MPPTs, a cellular flat radio, a GPS receiver (with flat antenna) and attitude sensors are adequate components to be integrated on such a panel. This concept is a hybrid of cellular, panel and integration concepts. An example is sketched in Figure 4.8. To differentiate from the nano-modular format as described in Section 4.3.2, this concept still assumes a standard internal envelope for payloads and internal stack units an only focuses on those components which are typically already exposed to the outer environment. When the electrical interfacing with the internal stack can be performed without loose wires, e.g. by the use of spring-loaded connectors, this concept allows a very easy and quick integration. Using as much as possible standard commercial-off-the-shelf electrical and mechanical components may introduce further economic advantages when production of these advanced panels can be fully automated similar to the production of consumer equipment.

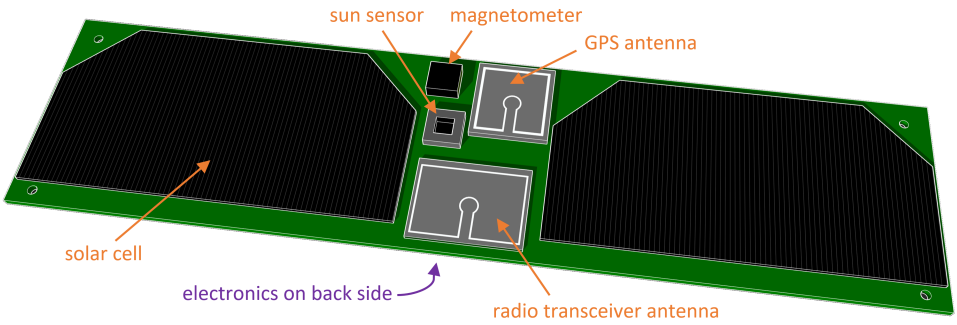


Figure 4.8: Artist impression of an advanced outer panel, suitable for a 3p PocketQube.

Such an advanced integrated outer panel would be most beneficial for very small satellites such as PocketQubes and small CubeSats, which would directly benefit from the easy assembly while the dimensions and tolerances are small enough to sustain the structural loads and making spring loaded connectors to the internal stack possible. On larger satellites, already with CubeSats beyond 2U, these panels require additional structural support and potentially flexible wiring harness to the inside. However, one could also consider to make such a panel a self-powered wireless unit for larger satellites.

4.4.6. Core Integrated Stack Unit

Integration of functions of a satellite on a single printed circuit board is a simple but effective mean to reduce volume. The most advanced implementation of this con-

cept would be the integration of the complete satellite on a single PCB. This would be a good concept for vastly distributed networks, where initial investment cost can be distributed over the individual satellites as investigated for the PCBSat concept [23]. However, the reduced modularity provides less versatility to adopt the entire satellite system to mission specific needs. Therefore, it would make most sense to integrate subsystem functions which are almost always present on a satellite, which can be miniaturized and do not scale too much with missions specific needs and/or satellite configuration. Especially functions which can reside on integrated circuits are good candidates, while mechanical systems such as attitude actuators and propulsion are less suitable. Also components which are very configuration dependent (such as attitude sensors or solar cells) would not be the best candidates for system integration. A first step would be to integrate the central OBC with the main power conversion, monitoring and distribution on a single PCB. A battery system would still be separate as this one highly scales in volume with the required capacity. Also MPPT circuitry can consume a considerable amount of board space, but integration should be feasible on the same CubeSat board while for PocketQubes they need to be integrated with the solar panels themselves (see concept in section 4.3). As a next step, the micro-controller used for the OBC could in principle also be used to run the attitude determination and control algorithms. Or, if this is undesired, one could opt for a second micro-controller on the same board. A MEMS internal measurement unit and magnetometer could further complement the core integrated stack unit. However, as stated before, some sensors are better not integrated on this unit to avoid potential configuration conflicts. Attitude actuators are highly scalable with the satellite size, configuration and mission requirements and should therefore preferably be on different (modular/cellular) systems.

The concept of a core integrated stack unit clearly re-configures the physical subsystem boundaries and integrates several functions on a single board while splitting several virtual subsystems of different units which nowadays typically are integrated on a single PCB or integrated unit (like EPS & ADCS). It is expected that this concept could save the equivalent of at least one standard printed circuit board with standard electrical interface connector, so about 0.1U of a CubeSat or 0.2p of a PocketQube.

Another approach to reduce volume on CubeSats is to have several (internal) PocketQube units mounted on a CubeSat main board. This could especially be useful for systems which can benefit from further miniaturization of electrical circuits, for example by the use of system-on-chips for radio frequency technology, computation and sensor systems, as these systems have no strong relation to the scaling of the satellite or its mission resource requirements. For scalable components, such as amongst others batteries, boards with attitude actuators and a propulsion unit, this will not be very beneficial. In case of cellularization of these type of PocketQube components for CubeSats, a direct mounting of these components on a CubeSat board is more volume efficient than when using PocketQube boards as interface in between.

4.4.7. Concepts Selection and Reliability

From the advanced architectural concepts stated in the previous subsections, there is no superior one nor it is possible to formulate an ideal hybrid architecture which suits all types of missions. Some of the stated concepts are not completely compatible with each other and each concept has advantages and disadvantages. There is a high degree of subjectivity when trading these concepts and the weight of criteria may be different for various missions. For example, for vastly distributed networks of identical satellites, the time of integration of the satellites is more important than for a single satellite mission. The approach to reliability also plays an important role when selecting an appropriate architecture. A case study on an existing CubeSat can provide some insight on the potential impact of a hybrid architecture.

The reliability philosophy can be a dominant factor in the system complexity and the volume taken by bus subsystems. In a subsystem without redundancy, Fault Detection, Isolation and Recovery (FDIR) mechanisms can be implemented to mitigate many faults and avoid that they become a potential failure. In software, FDIR can help to recover from undefined states of the satellite (subsystems). In the hardware, FDIR can prevent damage at latch-ups by quickly detecting overcurrents and subsequently power down the subsystem for a while. To be able to provide FDIR for permanent hardware failures, however, redundancy needs to be applied. Redundancy can be implemented by addition of components or full subsystems of the same design or by alternative backup systems of a different design. This requires more volume and more complex FDIR, since arbitration should be added while limiting the risk for false triggers and avoiding that the FDIR circuitry itself becomes a single-point-of-failure. In Section 4.2, it was already stated that making a single-point-of-failure free design by either multiplicative redundancy or alternative backup systems was very complex and time consuming for previous Delfi satellites. The Delfi-n3Xt satellite has been chosen as case study provided in Section 4.5.

4.5. Delfi-n3Xt Case Study

Delfi-n3Xt has a modular architecture similar to many other CubeSats combined with several reliability measures including full subsystem redundancy. It is therefore considered to be a suitable satellite to act as a limited case study on the potential impact of reliability measures and key architectural concepts. First the architecture of Delfi-n3Xt will be explained. Subsequently an attempt is made to apply various advanced architectural concepts on the Delfi-n3Xt satellite and assess the impacts of these theoretical configurations on the satellite's volume budget.

Delfi-n3Xt is a 3U CubeSat mission with education and technology demonstration as main objectives. The physical breakdown is shown in Figure 4.9. The electrical architecture of Delfi-n3Xt is provided in Figure 4.10. It shows key data and power interfaces between physical subsystem units, electronic circuits and a selection of components. A single-point-of-failure free design philosophy was initially applied to the critical subsystems for the key mission objectives. This yields redundancy of the On-Board Computer (OBC), the radio transceivers and parts of the electrical power subsystem. Only the radio transceivers have different physical units and

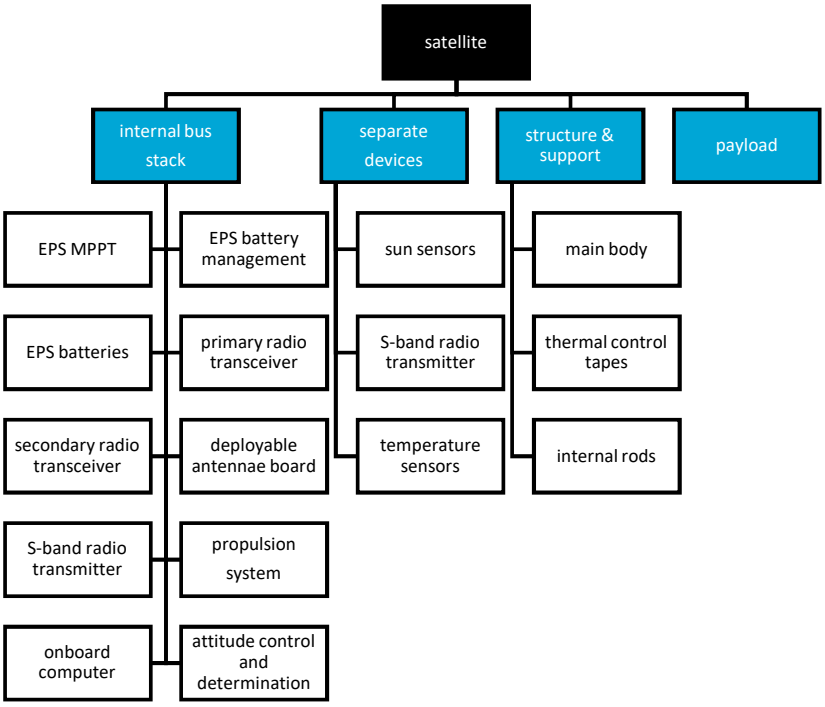


Figure 4.9: Delfi-n3Xt launch configuration physical breakdown.

can be considered fully redundant. On the OBC and EPS, the same PCB is used but the electrical circuits are fully isolated. Full three axis attitude determination and control is one of the technology demonstration subsystems and comprises a dedicated micro-controller, sun sensors, magnetometers, reaction wheels and magnetorquers. De-tumbling of the ADCS is considered mission critical and is secured with a simplified secondary circuit using a simple micro-controller and its own three-axis magnetometer. The three magnetorquers, including separate driving circuits for each of them, are placed on a separate board and connected to the main bus. Deployable antennae are mounted on a board which is attached via coaxial cables to the primary and secondary transceivers. They have redundant deployment circuitry and thermal knifes as well as electrical isolation circuits to avoid common mode failures on the radio transceivers. The propulsion system and the high-speed S-band transmitter are additional technology demonstration subsystems which are not critical to the rest of the satellite. They were however intended to become part of the critical bus for future missions once proven to function correctly. Only the solar cell experiment is truly a standalone payload.

All subsystems are equipped with micro-controllers which are connected to the main I^2C data bus, a linear data bus explained in Chapter 3. Components with digital interfaces, either on the same PCB or via dedicated wiring (e.g. sun sensors), use their own point-to-point data bus.

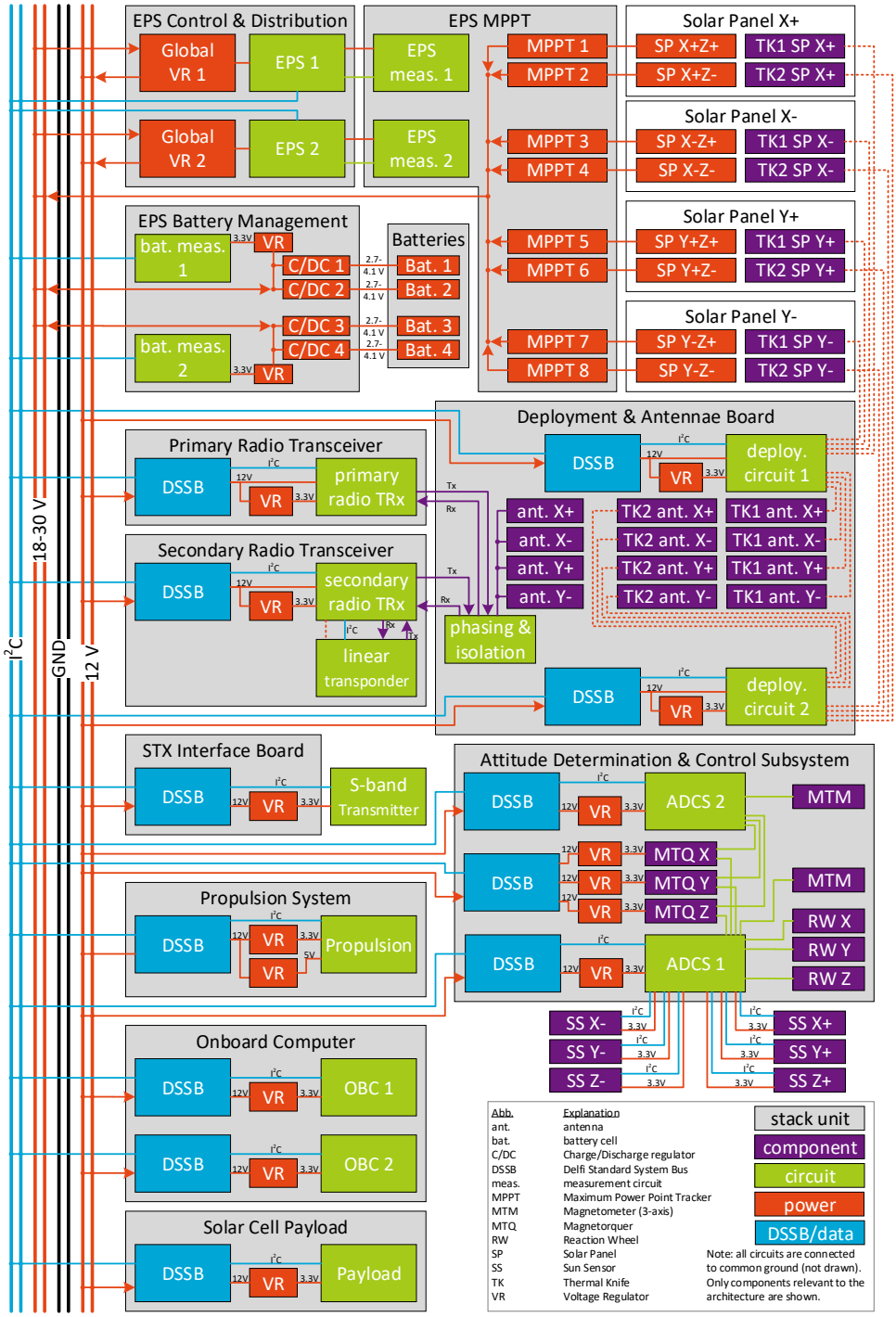


Figure 4.10: Top-level electrical architecture of Delfi-n3Xt as launched.

The electrical power subsystem provides a mix of full and cellular redundancy. There are eight separate Maximum Power Point Trackers (MPPTs) which are connected to the respective solar arrays. Failure of one string will still yield sufficient power for all mission objectives, so this is both a cellularization approach as well as redundancy. The MPPTs provide power to the distribution bus of variable voltage (18-30 V). A separate board is directly stacked on the EPS MPPT board. This board comprises two redundant micro-controllers and regulators. The micro-controllers are used to perform data acquisition and slow control of both boards, while the regulators convert the distribution bus to a regulated 12 V bus which connects all subsystems. At the subsystems, local DC-DC converters are used to regulate the power to the required voltages (typically 3.3 V and 5 V). A separate battery system comprises four batteries and connects to a battery management board which has four separate charge and discharge circuits and two redundant micro-controllers, which either charge the battery when the variable voltage bus is high or vice versa. The variable voltage bus has a significantly higher voltage than the individual battery cells and thus requires DC-DC conversion. There can be up to five DC-DC conversion steps in series from the solar panels to the subsystem components, which can yield up to 50% energy loss, which is discussed in Section 2.5.1.

The electrical interfaces between subsystems are not made fully redundant, except for the wiring lines. Protection circuitry was implemented in an attempt to make the system fail-safe. The Delfi Standard System Bus (DSSB) comprises key electrical interfaces and a circuit that is implemented on each physical subsystem unit or payload. Its circuit diagram is provided in Figure 4.11. The DSSB circuit can detect over-current of the local subsystem. It can also detect a loss of communication on the I^2C data bus. In sequential order, dependent on its critical functionality, subsystems are first reset and after a few attempts isolated by this circuit, waiting further instructions from the OBC. The wiring harness uses a flex-rigid PCB and Harwin Datamate connectors as shown in Figure 4.1. The rationale for choosing a flex-rigid PCB is to mechanically decouple the different subsystem units for an increase of the thermo-mechanical cycle life time. The potential problem with rigid stackable connectors was however never fully investigated, nor is it proven that a flex-rigid PCB is more reliable in this sense. The flex-rigid PCB requires a minimum distance between connected boards, which in a few cases yields empty and unusable volume between subsystems of limited height.

The DSSB design did unfortunately not result in a single point of failure free design since the DSSB circuit, used to protect the power busses and I^2C data bus, can still result in a failure at the main bus because the used commercial components for the DSSB do not inherently fail to high impedance state at the main bus side. This issue was discovered very late in the development due to a testing accident in which a temporary over-voltage on the subsystem was applied, causing one of the I^2C buffers of the DSSB to short the main side of the bus permanently. Due to lack of time left, this risk had to be accepted. Moreover, the redundancy concept and DSSB design is very complex and has consumed several man-years to implement. In the final phase of the development, there was only limited time left to test out all subsystems. A second issue is that the redundancy of critical (parts of) and the

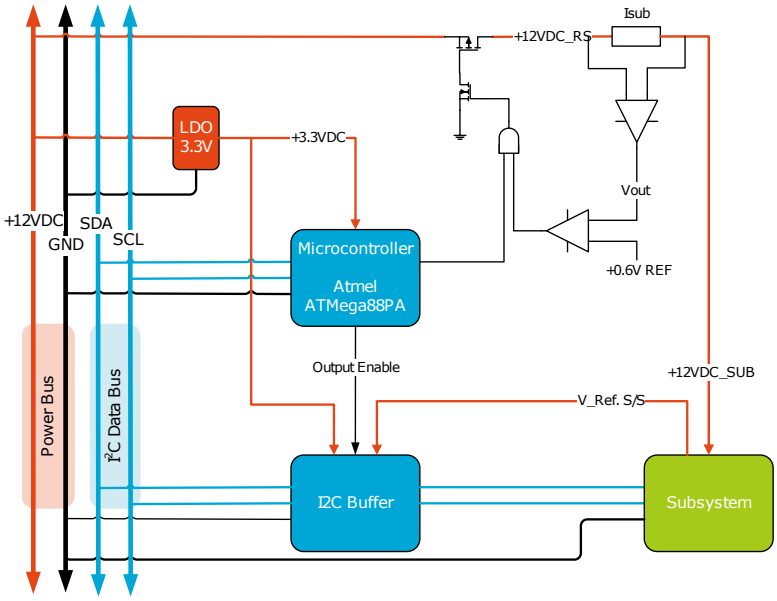


Figure 4.11: Electrical diagram of Delfi Standard System Bus circuit.

DSSB circuitry consumes considerable volume. From a reliability and complexity perspective, full redundancy of critical devices and cross-coupling their interfaces, as performed on the FLP satellite [24], would be a more solid approach and may even be less complex. In the case of more advanced missions, however, more subsystems would need to become fully redundant just like the electrical interfaces and the wiring harness. This would not have fitted in the 3U volume of a CubeSat.

Delfi-n3Xt has achieved primary mission success after three months in-orbit, after which is became silent for seven years. In February 2021, the satellite spontaneously restarted its transmission again. The causes for both event still remain unknown, but it can be concluded that the complex reliability approach using different redundancy concepts did not provide the reliability gain which it was aimed for. The volume consumption, complexity and the lack of resources to properly test out the satellite at system level, including its FDIR, are key reasons to aim at an improvement of reliability without redundancy of complete subsystems. In Chapter 5 this strategy will be further investigated. In the remainder of this section, the effect on the payload volume for a lean configuration without (full) redundancy and an advanced architecture will be studied.

In a hypothetical lean configuration variant of Delfi-n3Xt, all redundant systems would be removed. The S-band radio transmitter uses the OBC for wiring interface instead of a dedicated board. All spacing in between the units have been removed because of the use of the stack-able CS14 connector (see Section 3.6), which also results in reduction of height of the OBC and EPS boards and the inter-spacing of units overall.

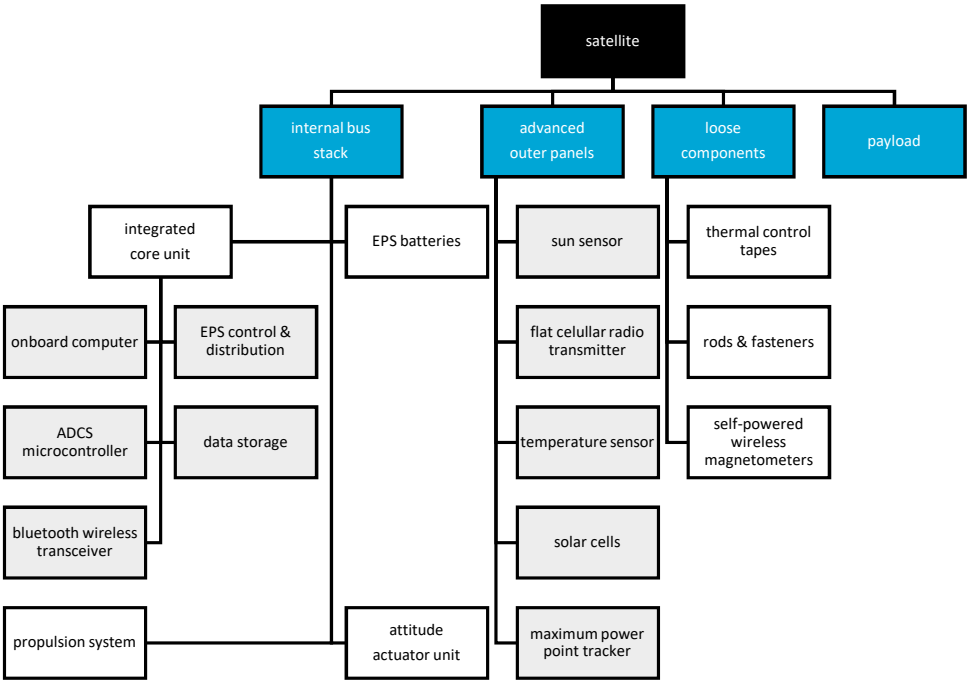


Figure 4.12: Delfi-n3Xt advanced concept physical breakdown.

In the advanced configuration, as shown in Figure 4.12 where grey boxes represent integrated components, an integrated core unit combines the EPS control and distribution, the OBC and the ADCS micro-controller. There is a separate attitude actuator board, which is slightly smaller than the previous ADCS system. The battery system and propulsion system remain unchanged. MPPT, sun sensors and flat cellular radios are integrated together with the solar cells on an advanced outer panel. Magnetometers are distributed over the satellite as self-powered wireless sensors.

The volume budgets of the different internal stack configurations are compared in Figure 4.13. The effective payload volume for all configurations is based on an internal volume of 90 mm x 90 mm square. The available payload stack height is 27 mm in the launch configuration, 165 mm in the lean configuration and 260 mm for the advanced configuration. This proves that a significant gain can be achieved in the available payload volume with a lean approach and a dramatic improvement with an advanced architecture.

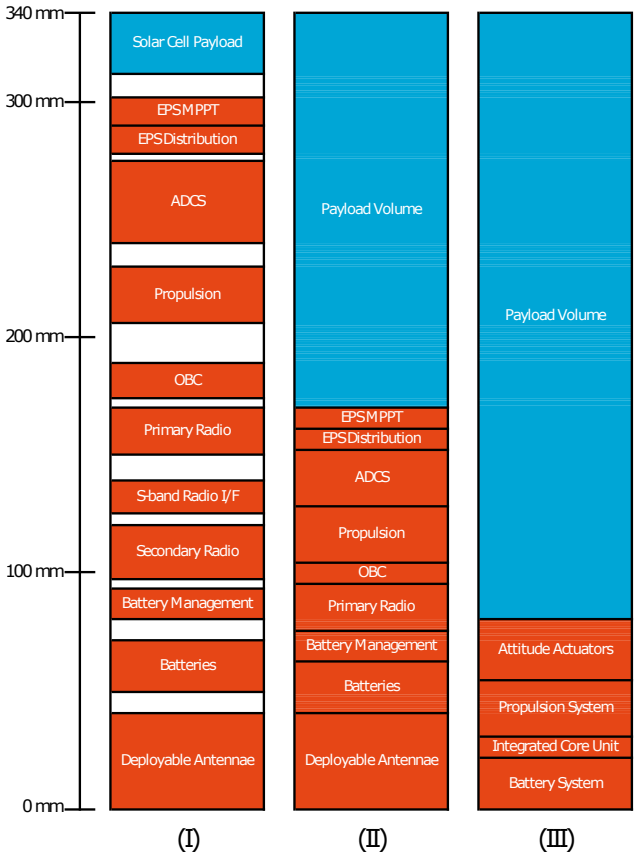


Figure 4.13: Delfi-n3Xt volume budget with launch (I), lean (II) and advanced (III) configuration.

4.6. Conclusions & Discussion

In this chapter, several traditional and advanced approaches with respect to the physical architecture of PocketQubes and CubeSats have been presented and analyzed theoretically. Cellularization of components, integration of core subsystem functionality into a single physical unit, an advanced outer panel and self-powered wireless sensors are all advanced and promising concepts. For each of them advantages and disadvantages compared to a typical modular approach found in CubeSats have been analysed. In particular, it can be concluded that the advanced concepts typically become impractical when implemented in a dogmatic way for the whole satellite and therefore a smart pragmatic approach is recommended. A hybrid approach, using a mix of the traditional approach with advanced concepts can be very good compromise, but it should be noted that some concepts are not fully compatible with each other. Plug-and-play is an interesting but not yet mature concept. A lean electrical interface standard, such as proposed in Section 4.3.4, can be implemented independently on the short term, while the development of plug-and-play

can focus purely on the software implementation.

With respect to reliability, it is argued that a dogmatic redundancy approach is counter-effective within the resource-limited environment (both technical as well as organizational) of CubeSats and PocketQubes. Satellite developers are recommended to start first with a satellite using single subsystem and make this as reliable as possible before adding additional reliability features such as redundancy. Overall, a more pragmatic approach would be advised in which only components which are wearing out mechanically (e.g. reaction wheels) or due to cycling (e.g. battery cells) should be addressed by (over-dimensioned) cellularization and/or multiplicative redundancy. However, it should be noted that this recommended approach is in contrast with the conclusion in the reference on the artificial stem cells [12] (see Section 4.3.1). Further investigation in the reliability of CubeSats and PocketQubes and the impact of redundancy will be performed in Chapter 5.

When a lean electrical interface standard is implemented and full system redundancy is omitted, significant payload volume can be achieved. With a Delfin3Xt case study, it is shown that such a simple step would increase the payload volume to about nearly half of the internal 3U CubeSat volume. When using an advanced approach by integrating some core satellite functions on a single internal PCB and re-allocation some circuits and components to advanced outer panels, one can even increase this to three-quarters of the internal volume while gaining reliability through cellularization of some components.

A follow up of this study would be to perform laboratory testing and in-orbit demonstration of several concepts. Reliability of the concepts should be investigated further to validate that full subsystem redundancy is not the ideal approach to increase the reliability of CubeSats and PocketQubes. Likewise, this analysis is needed in order to compare the advanced concepts to a traditional modular approach. If the reliability does not become a major issue, the advanced architectural concepts presented have potential to become the new norm for very small satellites.

References

[1] J. Bouwmeester, E. Gill, S. Speretta, and S. Uludag, *A New Approach on the Physical Architecture of CubeSats & PocketQubes,* Journal of the Britisch Interplanetary Society **71**, 1 (2018).

[2] D. Geeroms, S. Bertho, M. De Roeve, R. Lempens, M. Ordies, and J. Prooth, *ARDUSAT, an Arduino-based CubeSat Providing Students with the Opportunity to Create Their Own Satellite Experiment and Collect Real-World Space Data,* in *Proceedings of the 22nd ESA Symposium on European Rocket and Balloon Programes and Related Research* (ESA, Tromso, 2015).

[3] A. Aslan, M. Bas, S. Uludag, S. Turkoglu, and Et.al., *The Integration and Testing of BeEagleSat,* in *Proceedings of the 30th International Symposium on Space Technology and Science (ISTS), the 34th International Electric Propulsion Conference (IEPC) & the 6th Nano-Satellite Symposium (NSAT)* (Kobe, 2015).

[4] H. Ehrpais, I. Sünter, E. Ilbis, J. Dalbins, I. Iakubivskyi, E. Kulu, I. Ploom, P. Janhunen, J. Kuusk, J. Sate, R. Trops, and A. Slavinskis, *ESTCube-2 Mission and Satellite Design,* in *Proceedings of the 4S Symposium* (ESA, Valetta, 2016).

[5] L. Sha, M. Koenraad, S. W. Seng, G. C. Hiang, and A. Ling, *Galassia System and Mission,* in *Proceedings of the 28th AIAA/USU Conference on Small Satellites*, edited by AIAA (Logon, 2014).

[6] L. Alminde, M. P. I. Busgaard, D. Smith, and L. Perez, *GOMX-4: Demonstrating the Building Blocks of Constellations,* in *Proceedings of the 31st AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2016).

[7] J. Bouwmeester, G. Aalbers, and W. Ubbels, *Preliminary Mission Results and Project Evaluation of the Delfi-C3 Nano-Satellite,* in *Proceedings of the 4S Symposium* (ESA, Rhodes, 2008).

[8] J. Bouwmeester, L. Rotthier, C. Schuurbiers, W. T. Wieling, G. V. D. Horn, F. Stelwagen, E. Timmer, and M. Tijssen, *Preliminary Results of the Delfi-n3Xt Mission,* in *Proceedings of the 4S Symposium* (ESA, Mallorca, 2014).

[9] *WREN: A HAM radio SSTV PocketQube,* (2013).

[10] D. Cachia, J. Camilleri, M. A. Azzopardi, M. Angling, and A. Sammut, *Feasibility Study of a PocketQube Platform to Host an Ionospheric Impedance Probe,* in *Proceedings of the 4S Symposium* (ESA, Valetta, 2016).

[11] T. Jaeger, W. Mirczak, and B. Crandall, *Cellularized Satellites - A Small Satellite Instantiation that Provides Mission and Space Access Adaptability,* in *Proceedings of the 31st AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2016).

[12] A. O. Erlank and C. P. Bridges, *Satellite Stem Cells: The Benefits & Overheads of Reliable, Multicellular Architectures,* in *IEEE Aerospace Conference Proceedings* (IEEE, Big Sky, 2017).

[13] C. McNutt, R. Vick, H. Whiting, and J. Lyke, *Modular Nanosatellites - Plug-and-Play (PnP) CubeSat,* in *Proceedings of the 7th Responsive Space Conference* (AIAA, Los Angeles, 2009).

[14] C. J. Kief, B. K. Zufelt, J. H. Christensen, and J. K. Mee, *Trailblazer: Proof of Concept CubeSat Mission for SPA-1,* in *Proceedings of AIAA Infotech* (AIAA, St. Louis, 2011).

[15] F. Bruhn, J. Schulte, and J. Freyer, *Njord: A Plug-and-Play Based Fault Tolerant CubeSat Architecture,* in *Proceedings of the 4S symposium* (ESA, Portoroz, 2012).

[16] J. Lyke, J. Mee, F. Bruhn, G. Chosson, R. Lindegren, H. Lofgren, J. Schulte, S. Cannon, J. Christensen, B. Hansen, R. Vick, and J. Calixte-Rosengren, *A Plug-and-play Approach Based on the I2C Standard,* in *Proceedings of the 24th AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2010).

[17] J. Lyke, Q. Young, J. Christensen, and D. Anderson, *Lessons Learned : Our Decade in Plug-and-play for Spacecraft,* in *Proceedings of the AIAA/USU Conference on Small Satellites* (AIAA, 2014).

[18] C. W. de Boom, N. van der Heiden, J. Sandhu, H. C. Hakkesteegt, J. L. Leijtens, L. Nicollet, J. Bouwmeester, G. van Craen, S. Santandrea, F. Hannoteau, C. W. D. Boom, N. V. D. Heiden, J. Sandhu, H. C. Hakkesteegt, J. L. Leijtens, L. Nicollet, J. Bouwmeester, G. V. Craen, S. Santandrea, and F. Hannoteau, *In-Orbit Experience of TNO Sun Sensors,* in *Proceedings of the 8th International Conference on Guidance, Navigation and Control* (ESA, Karlovy Vary, 2011).

[19] J. Llanos and J. Bouwmeester, *Thermoelectric Harvesting for an Autonomous Self-Powered Temperature Sensor in Small Satellites,* in *Proceedings of the 68th International Astronautical Congress* (IAF, Adelaide, 2017).

[20] J. Bouwmeester, J. Reijneveld, T. Hoevenaars, and D. Choukroun, *Design and Verification of a Very Compact and Versatile Attitude Determination and Control System for the Delfi-n3Xt Nanosatellite,* in *Proceedings of the 4S Symposium* (ESA, Portoroz, 2012).

[21] T. Vergoossen, J. Guo, J. Bouwmeester, and W. A. Groen, *Design, Integration, and Testing of World's Smallest Satellite Reaction Wheel,* in *Proceedings of the International Astronautical Congress, IAC* (2017).

[22] GOMspace, *NanoPower P31 (rev. 2.20)*, Tech. Rep. (2017).

[23] D. Barnhart, T. Vladimirova, M. Sweeting, R. Balthazor, L. Enloe, H. Krause, T. Lawrence, M. Mcharg, J. Lyke, J. White, and A. Baker, *Enabling Space Sensor Networks with PCBSat,* in *Proceedings of the AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2007).

[24] J. Eickhoff, *The FLP Microsatellite* (Springer, 2016).

5

Satellite and Subsystem Reliability of CubeSats

All models are wrong,
but some are useful.

George Box

I have not failed,
but found 1000 ways to not make a light bulb.

Thomas Edison

The objective of this chapter is to investigate which approach would lead to more reliable CubeSats: full subsystem redundancy or improved testing. Based on data from surveys, the reliability of satellites and subsystems is estimated using a Kaplan-Meier estimator. Subsequently, a variety of reliability models is defined and their maximum likelihood estimates are compared. A product of a Lognormal distribution addressing immaturity failure and a Gompertz distribution addressing wear-out is found to best represent CubeSat reliability. Bayesian inference is used to find realistic wear-out parameters by using failure data of small satellites. Subsystem reliability estimates are subsequently found using a similar approach. A reliability model for CubeSats with redundant subsystems is established, verified and applied in a Monte Carlo simulation. The results are compared with a model for reduced immaturity failure. Reduction of immaturity failures through improved testing is considered to be superior to subsystem redundancy considering limited resources.

Parts of this chapter have been submitted to the journal Reliability Engineering and System Safety.

5.1. Analysis Objective, Method and Scope

5.1.1. Objective for Reliability Modelling

The investigation and design choices of the proposed new electrical interface standard for PocketQubes and CubeSats are based on non-redundant subsystems and interfaces. While the proposed interface standard in Chapter 3 is not prohibitive for subsystem redundancy, the interfaces themselves remain non-redundant and the number of power distribution lines (4 for PocketQubes and 8 for CubeSats) and potential data bus nodes limits the possibility of subsystem redundancy. The innovative satellite architecture, proposed in Section 4.6, also favours volume reduction of the core satellite bus for additional payload volume, resulting in a system without full subsystem redundancy. However, as 37% of CubeSat missions fail within the first year of operations, reliability is a major concern and applying redundancy is a commonly applied measure to improve system reliability. The potential of subsystem redundancy is therefore considered to be a driving architectural aspect for reliability. The implicit hypothesis to exclude redundancy in the proposed architecture and electrical interfaces is that the required financial, technical and human resources, which are needed for implementing redundancy, can more effectively be allocated to improvement of the reliability of the individual subsystems and interfaces through more extensive testing and measures to mitigate potential failures. This is considered to be a driving development aspect for reliability. The original research question can thus be made specific and becomes:

What leads to higher CubeSat reliability over its mission life time: full subsystem redundancy or improved testing?

Resources, otherwise spent on implementing redundancy, can be also used for operating more satellites in a distributed network for a certain application. This concept was already investigated by Engelen et al. [1], who have shown that a satellite swarm with a few redundant satellites can have a high reliability even if the reliability of a single satellite is limited. However, Engelen et al. [1] assume a lower infant mortality compared to those identified in another study on CubeSats [2]. This assumption is justified if the CubeSat design is very mature, which can be achieved by implementing predecessor missions to test and improve a specific satellite design iteratively. For individual satellites or small satellite networks 'swarm robustness' [1] does not apply. To investigate the reliability for all types of CubeSats missions, a new modelling approach is required.

5.1.2. Methodology

The steps which are taken to answer the research question presented in Section 5.1.1 are provided in 5.1. The method applied in this chapter is a mathematical simulation based on empirical CubeSat reliability data. The best metric to answer the research question is: reliability over time in orbit.

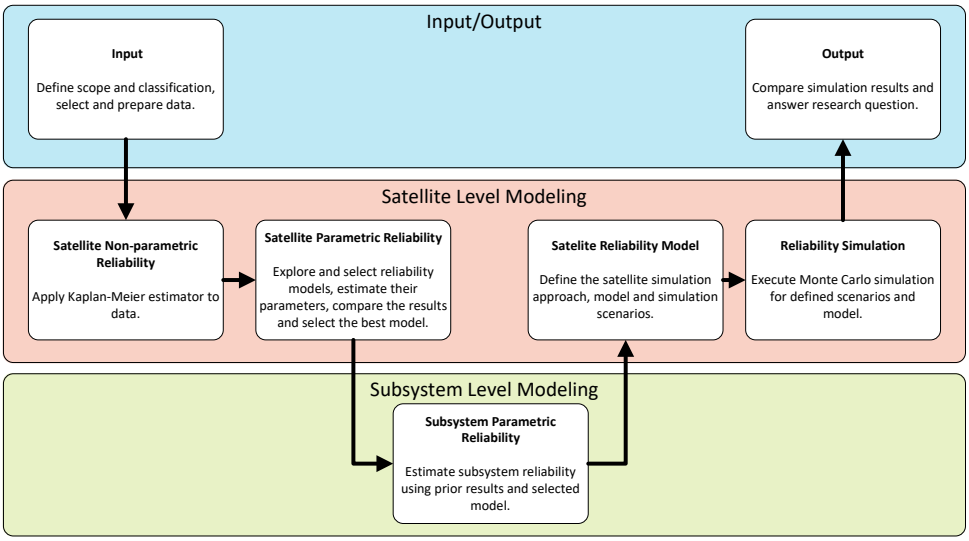


Figure 5.1: Flow diagram of reliability analysis applied in this chapter.

Scope of Study

The scope of this reliability study is limited to catastrophic and irrecoverable critical failures of CubeSats caused by failing subsystems including their physical interfaces. This means that a failure within a subsystem causes loss of the satellite or its mission [3]. All minor, major and recoverable failures which do not cause the satellite or mission to fail entirely are ignored in this study as they are not directly related to the research question concerning full subsystem redundancy. CubeSats are considered to have a limited mission scope for which typically all subsystems are essential to function properly. If a critical subsystem is not able to recover from its failure, only redundancy can help to mitigate the risk of satellite failure. For modelling the risk of a satellite within the defined scope, investigating the effect of subsystem redundancy and improved testing, a model for irrecoverable subsystems failures is needed.

5.2. Classification and Selection of Failure Data

5.2.1. Failure Classification

There are many possible ways to classify the root causes of failures. For this study, a classification is desired which has the ability to model the failure rate over time continuously. The bathtub curve is a widely used theoretical reliability model for the failure rate of a group of devices. Its origin is unknown, but it is described many times for hundreds of years [4]. An example of the bathtub curve is shown in Figure 5.2.

The bathtub curve starts at the roll-out of operations of a system with a declining failure rate. Failures due to poor design, production errors of components, too

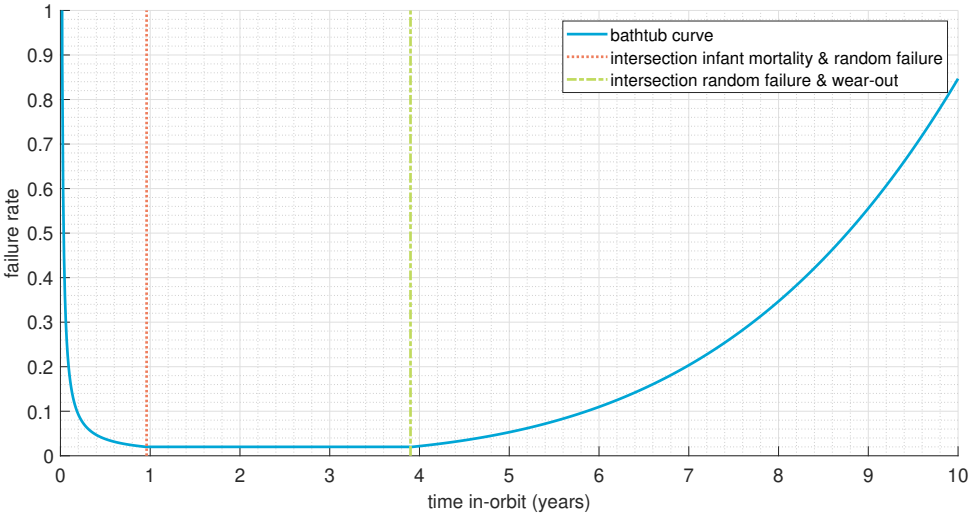


Figure 5.2: Schematic bathtub failure rate curve.

limited testing and/or wrong analysis of the operational environment may lead to early failure in life. For this reason, these failures are often called 'infant mortality'. Subsequently follows a period of random failures, represented by a constant failure rate. At a later lifetime of the system, wear-out failures are shown with increasing failure rate. Eventually every satellite subsystem will wear-out due to accumulation of environmental effects and/or internal ageing throughout its active operation. Wear-out is thus inevitable on the long term irrespective of the maturity of the development.

Although the bathtub curve in Figure 5.2 provides a continuous model over time, it has a few issues. When using adjacent time windows, each class of failures is cut-off in time. This is not realistic for any of the failure classes. Moreover, the required boundary conditions, to avoid discontinuities in failure rate, will complicate the estimation of model parameters. A compound of continuous models over time, instead of adjacent, is therefore preferred. In this case, however, the tails of the decreasing and increasing failure rate models may be difficult to distinguish from a constant failure rate model. Furthermore, according to Klutke et al., there is a lack of empirical evidence for the theoretical bathtub curve [4] and according to Wong 'there is no such thing as random failure' [5]. If a specific random event occurs frequently, e.g. many times per day, the majority of satellites are developed such that they are able to survive this. However, some may fail. If a specific random event occurs early in life, the term 'infant mortality' applies. If the average time between such random events increases, the potential failure rate will decrease and the period in which a failure can be expected increases. This makes the term infant mortality less suitable. There are many different events which may cause failure and the time between these events varies. The events may be systematic (e.g. thermal cycling in orbit), random (e.g. particle radiation), deterministic (e.g. a

software state) or user-imposed (e.g. change of operation by telecommand). All those events together will yield a continuously decreasing failure rate and not a constant one. Extensive Failure Mode and Effect Analysis (FMEA), adequate design and extensive on-ground and in-orbit testing reduces the risk for these types of failures and increases the maturity of the satellite. Therefore, in this work, the term 'immaturity failure' is introduced which comprises infant mortality as well as any other random, deterministic or operations-induced single event failures. Immaturity failures and wear-out failures together constitute all possible satellite failures.

5.2.2. Available Failure Data

The data which is selected for this study is the 'CubeSat failure database'. This database originates from the results of the same survey which is introduced in Section 2.1 and is further extended through literature search and individual correspondence by Langer [2]. The database contains 71 observed failures from 178 CubeSats for an observation window from 20 May 2003 to 31 December 2014. It contains censor times and failure times at satellite and subsystem level. A failure indicates the loss of satellite operability or loss of a main mission objective. A censor time indicates that the satellite was turned off intentionally after achieving mission success, de-orbited after successful operations or was still operational at the time of inquiry. A histogram of the data is shown in Figure 5.3. The original CubeSat failure database, used in Section 2.1, also contains some information on mission design life times. They range from one month to five years for launched CubeSats.

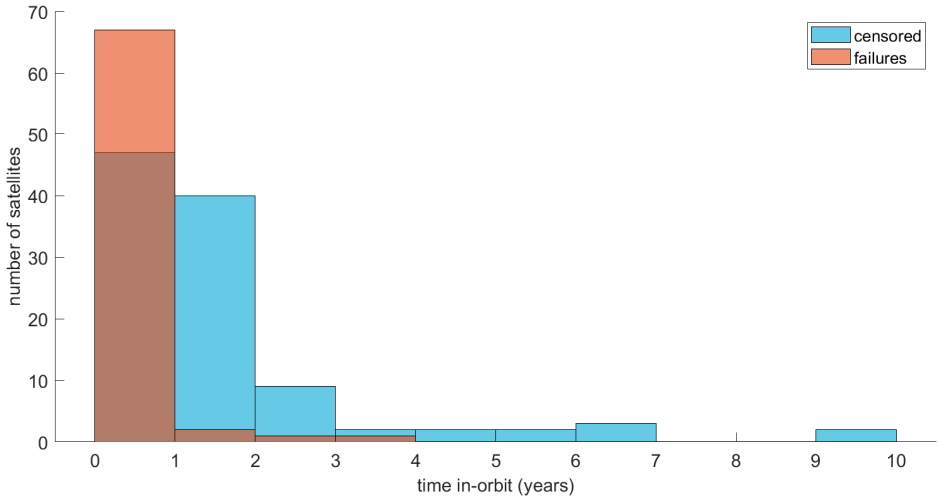


Figure 5.3: Histogram of data from CubeSat failure database.

A second database containing small satellite failures data, already used in another study at TU Delft by Guo and Monas [6], is also available for this study. It contains data on 152 satellites launched between 1990 and 2010 with a mass lower than 500 kg and reports 83 failures and 69 censored items. The small satel-

lite database will be used to check whether the model selection for CubeSat failures can be generalized to other classes of satellites. Furthermore, the fact that in this database there are relatively more failures beyond one year compared to the CubeSat failure database, as can be seen in Figure 5.4, will be used as input for Bayesian inference as explained in Section 5.5.4.

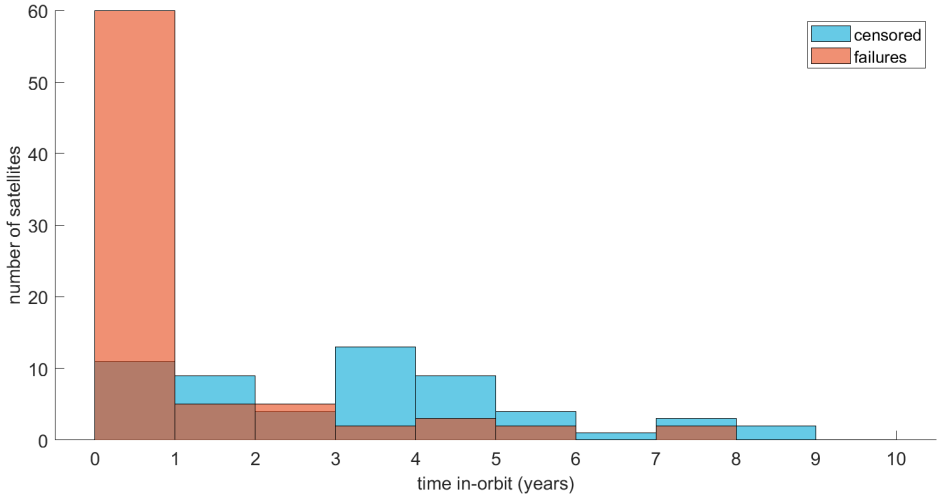


Figure 5.4: Histogram of data from small satellite failure database.

5.3. Non-parametric Reliability Model

Observation of failures from a population of orbiting satellites can be used to estimate their reliability over time. A typical problem with the data of satellite failures from a heterogeneous population is that satellites might have been retired without a failure or satellites are still operational at the observation date of the survey. The first step of the reliability analysis is therefore to censor this data. The Kaplan Meier Estimator (KME) [7] is a non-parametric survival function ($\hat{S}$) which can be directly established from failure and censor times of satellite and subsystem data [8, 9].

$$\hat{S}(t) = \prod_{i:t_i<t} \left(1 - \frac{d_i}{n_i}\right) \tag{5.1}$$

KME provides the estimated survival function over time t and is updated at each time t_i that a number of failures d_i occur, having n_i operational units at risk. If the time resolution becomes infinitesimal, d_i will equal one for satellites which fail independent of each other. In the databases, the time resolution is expressed in days which leads to several occasions where $d_i > 1$. The number of operational units at risk n_i indicates the total number of satellites minus all failed and censored devices up to the failure event time t_i. The variance of the estimator can be calculated by Greenwood's formula [10]. The confidence interval (CI) can subsequently

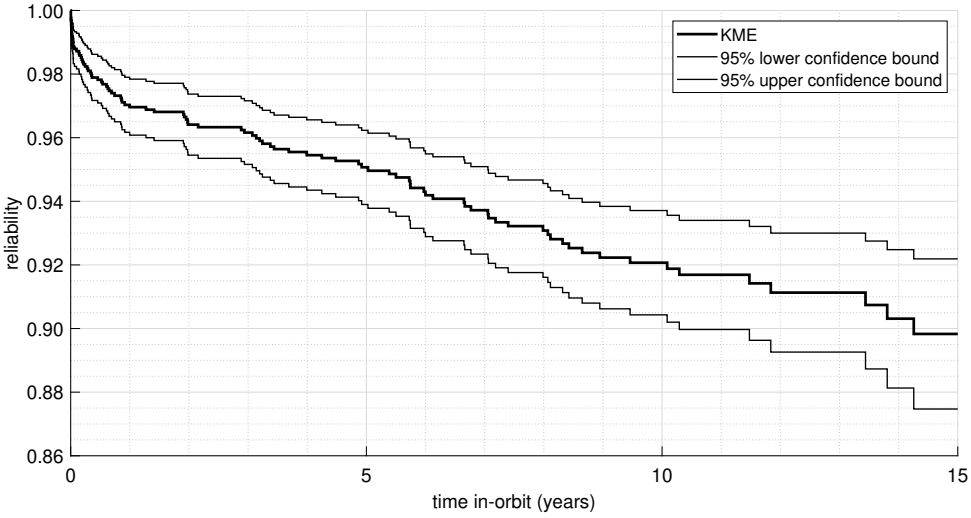


Figure 5.5: KME for satellites based on the data from [11].

be determined by applying the α-quantile of the normal distribution $z_{\alpha/2}$. For a 95% confidence interval, $z_{\alpha/2} = z_{0.025} = 1.96$.

$$\text{var}\left(\hat{S}(t)\right) = \hat{S}(t)^2 \sum_{i:t_i<t} \left(\frac{d_i}{n_i(n_i - d_i)}\right) \tag{5.2}$$

$$\text{CI} = \hat{S}(t) \pm z_{\alpha/2}\sqrt{\text{var}\left(\hat{S}(t)\right)} \tag{5.3}$$

An example of the KME is provided in Figure 5.5, using data of 1584 satellites launched between 1990 and 2008 [11]. A non-parametric model, such as the KME, provides a simple tool to visualize the reliability of a population over time. It is an exact representation of the empirical data and is therefore also unbiased and without assumptions. In the absence of failures over a period of time however, even if there are censored items, KME leads to a constant reliability value as can be seen by the horizontal lines in the example provided by Figure 5.5. While this is not a major issue when interpreting or sampling directly from the reliability function, it cannot be converted directly into failure rate or failure probability density. Moreover, KME does not allow differentiation in failure classes. A continuous parametric distribution does not exhibit these issues. KME is still a useful first step as it can act as input for a least squares estimator of parametric models and as a reference to assess the goodness-of-fit of a parametric model.

5.4. Parametric Reliability Models

In contrast to non-parametric models, parametric models provide a smooth distribution over time to 'fit' the empirical data. The basic metric is the reliability as

function of time $R(t)$, similar to $\hat{S}(t)$ of KME, representing the expected fraction of survivors over time. From this, other measures of reliability can be derived such as the probability of failure $F(t)$, the probability density $f(t)$ and the failure rate $\lambda(t)$ [12].

$$F(t) = 1 - R(t) \tag{5.4}$$

$$f(t) = \frac{dF(t)}{dt} = -\frac{dR(t)}{dt} \tag{5.5}$$

$$\lambda(t) = \frac{f(t)}{R(t)} \tag{5.6}$$

5.4.1. Basic Reliability Distribution Models

There are various candidates for parametric reliability distribution models. Previous research on satellite lifetime reliability has focused primarily on the Weibull distribution [2, 6, 11]. As indicated in Section 5.2, there is an interest in models capable of representing both immaturity failures and wear-out failures. This investigation is complemented with the gamma, Gompertz, log-logistic and log-normal distributions which are often applied in survival analysis in general [13, 14] to identify if they would yield better distributions than the Weibull.

Weibull Distribution

The Weibull distribution is a probability distribution with a wide range of applications [15], including the possibility to describe each failure class of the bathtub curve. Its reliability $R(t)$ is an exponential function with scale parameter θ, determining the time interval until 63.2% of the population have failed, and shape parameter β.

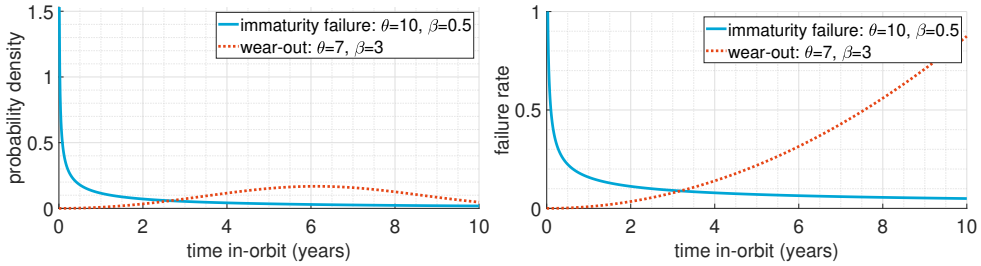
$$R(t) = \exp\left[-\left(\frac{t}{\theta}\right)^{\beta}\right] \tag{5.7}$$

Using Equations 5.5 and 5.6, the probability density function $f(t)$ and failure rate $\lambda(t)$ can be derived.

$$f(t) = \frac{\beta}{\theta}\left(\frac{t}{\theta}\right)^{\beta-1} \exp\left[-\left(\frac{t}{\theta}\right)^{\beta}\right] \tag{5.8}$$

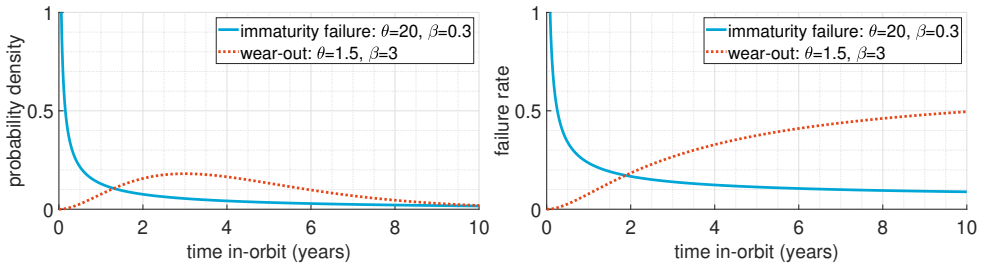
$$\lambda(t) = \frac{\beta}{\theta}\left(\frac{t}{\theta}\right)^{\beta-1} \tag{5.9}$$

For $\beta < 1$, the failure rate is monotonically decreasing over time, addressing 'immaturity failure'. For $\beta = 1$, the failure rate is constant, potentially addressing random failures. For $\beta > 1$, the failure rate monotonically increases. For $\beta > 2$ the probability density starts concave upward, which is considered a suitable boundary condition for wear-out phenomena. Examples are provided in Figure 5.6.



(a) probability density (b) failure rate

Figure 5.6: Examples of the Weibull distribution.



(a) probability density (b) failure rate

Figure 5.7: Examples of the gamma distribution.

Gamma Distribution

The gamma distribution is based on the gamma function, where Γ is the complete gamma function and γ the incomplete lower gamma function.

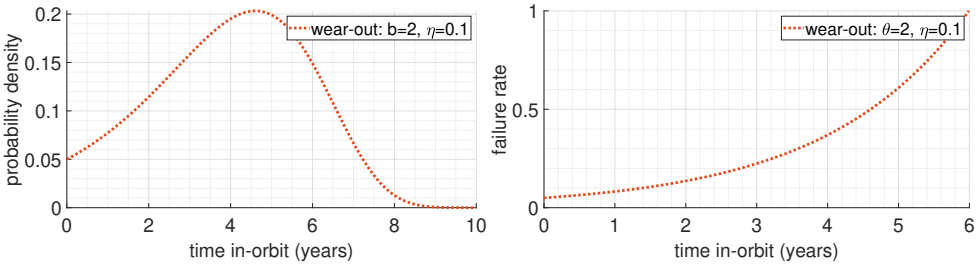
$$R(t) = 1 - \frac{\gamma(\beta, \frac{t}{\theta})}{\Gamma(\beta)} \tag{5.10}$$

$$\Gamma(\beta) = \int_0^\infty t^{\beta-1} e^{-t} dt \tag{5.11}$$

$$\gamma(\beta, \frac{t}{\theta}) = \int_0^{\frac{t}{\theta}} t^{\beta-1} e^{-t} dt \tag{5.12}$$

$$f(t) = \frac{t^{\beta-1} e^{-\frac{t}{\theta}}}{\Gamma(\beta)\theta^\beta} \tag{5.13}$$

Similar to the Weibull distribution, for $\beta \leq 1$ the failure rate is monotonically decreasing and for $\beta > 1$ it is increasing. The wear-out probability density of the Gamma distribution has a relatively long right side tail compared to Weibull as shown in Figure 5.7.



(a) probability density (b) failure rate

Figure 5.8: Examples of the Gompertz distribution.

Gompertz Distribution

Gompertz created a distribution for the lifetime of human beings, for which the death rate increases exponentially [16]. It uses shape parameter η and scale parameter θ.

$$R(t) = \exp\left[-\eta\left(e^{\left(\frac{t}{\theta}\right)} - 1\right)\right] \tag{5.14}$$

$$f(t) = \frac{\eta}{\theta} e^{\left(\frac{t}{\theta}\right)} \exp\left[-\eta\left(e^{\left(\frac{t}{\theta}\right)} - 1\right)\right] \tag{5.15}$$

$$\lambda(t) = \frac{\eta}{\theta} e^{\left(\frac{t}{\theta}\right)} \tag{5.16}$$

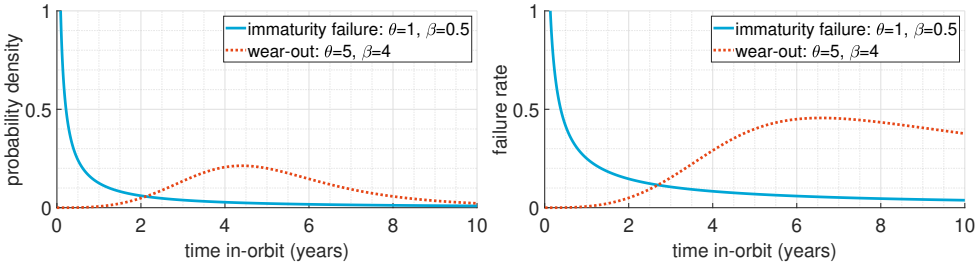
A different symbol is used for the shape factor as its behaviour is significantly different compared to Weibull and Gamma. The failure rate is always increasing for the Gompertz distribution, which makes it less suitable to address immaturity failure. For wear-out, the right side tail of the probability density is relatively short which makes it highly suitable for modelling human lifetime which is naturally bound. However, when $\eta \geq 0.1$, the probability density at $t = 0$ is already significant (>25% of the value at the mode) which would interfere with immaturity failure. In this case, a boundary condition of $\eta \leq 0.1$ is deemed appropriate.

Log-Logistic Distribution

The log-logistic distribution is used in survival analysis where the mortality rate first increases and later decreases. This model is applicable to some human diseases. This distribution can be used for immaturity failure with a monotonically decreasing failure rate with shape factor β<1. For wear-out of satellites it would be less ideal as there is no treatment for or recovery from wear.

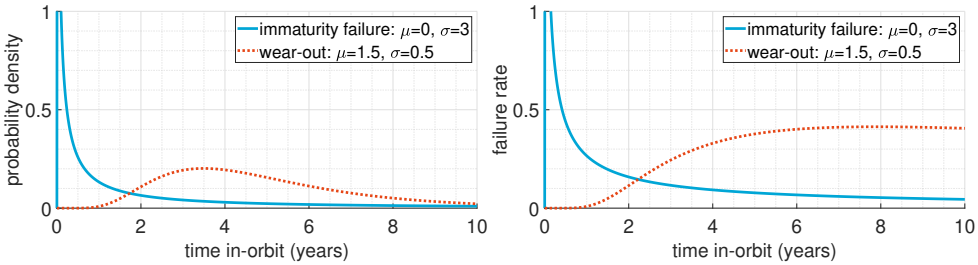
$$R(t) = 1 - \frac{1}{1 + (t/\theta)^{-\beta}} \tag{5.17}$$

$$f(t) = \frac{(\beta/\theta)(t/\theta)^{\beta-1})}{(1 + (t/\theta)^{\beta})^2} \tag{5.18}$$



(a) probability density (b) failure rate

Figure 5.9: Examples of the log-logistic distribution.



(a) probability density (b) failure rate

Figure 5.10: Examples of the log-normal distribution.

$$\lambda(t) = \frac{(\beta/\theta)(t/\theta)^{\beta-1})}{1+(t/\theta)^{\beta}} \tag{5.19}$$

Log-Normal Distribution

A normal distribution cannot be used for lifetime analysis as its range is $[-\infty, \infty]$. In a log-normal distribution, the logarithm of the random variable has a normal distribution and its range is $[0, \infty]$.

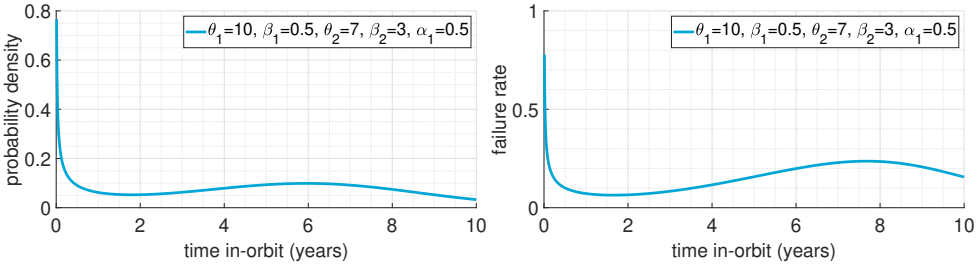
$$R(t) = 1 - \frac{1}{\sigma\sqrt{2\pi}} \int_0^t \frac{2}{x} \exp\left[-\frac{1}{2}\frac{(\ln(x)-\mu)^2}{2\sigma^2}\right] dx \tag{5.20}$$

$$f(t) = \frac{1}{t\sigma\sqrt{2\pi}} \exp\left[-\frac{(\ln(t)-\mu)^2}{2\sigma^2}\right] \tag{5.21}$$

This distribution cannot produce a monotonically decreasing failure rate for immaturity failure, nor a monotonically increasing failure rate for wear-out.

The distinction between immaturity failure and wear-out can be made using the mode Mo.

$$Mo = \exp(\mu - \sigma^2) \tag{5.22}$$



(a) probability density (b) failure rate

Figure 5.11: Example of Weibull mixture.

If $\sigma^2 >> \mu$, the mode can be skewed towards $t = 0$, which can be used to model immaturity failure. For wear-out, the log-normal distribution has a long right tail similar as for the gamma and log-logistic distributions.

5

5.4.2. Compound Reliability Distribution Models

The basic reliability distribution models presented in Section 5.4.1 can be used to model immaturity failure or wear-out, while a model is needed which is a compound of both. In this section, several ways to create a compound model of the basic models are presented.

Reliability Mixture Model

A mixture of n Weibull distributions is a common method applied in previous satellite reliability studies [2, 11] using α_i as the normalized weight factor for each component i.

$$R(t) = \sum_{i=1}^{n} \alpha_i R_i(t) \tag{5.23}$$

Using Equations 5.5 and 5.6, the probability density function $f(t)$ and failure rate $\lambda(t)$ can be derived. The number of parameters to be estimated is $3n - 1$.

$$f(t) = -\frac{d\left(\sum_{i=1}^{n} \alpha_i R_i(t)\right)}{dt} = \sum_{i=1}^{n} \alpha_i f_i(t) \tag{5.24}$$

$$\lambda(t) = \frac{\sum_{i=1}^{n} \alpha_i}{\sum_{i=1}^{n} \alpha_i R_i(t)} \tag{5.25}$$

An easily made logical error, as for example in reference [17], is to use the weighted sum of individual failure rates for the failure rate of a mixture. The failure rate is dominated by the relative fraction of survivors for each population, not the weight factor. A mixture model divides all devices in populations, where the weight factor α can be regarded as the probability of failing according to a basic model, for

example immaturity failure or wear-out. A potential problem may arise when there are relatively more survivors left for immaturity failure than for wear-out. This will, for example, occur for a Weibull mixture distribution due to the decreasing failure rate for immaturity failure and increasing failure rate for wear-out as shown at the right tail beyond approximately 8.5 years in Figure 5.11. This behaviour is not realistic. However, it will only be a problem if the fraction of survivors is still significant. It is considered acceptable if the fraction of survivors is less than 1% at the peak of the failure rate. This needs to be checked after the estimation of the parameters.

Castet & Saleh have applied the Weibull-Weibull mixture in their study [11] which was introduced in Section 5.3. They use the maximum error and average error over time between the Weibull distribution and the non-parametric data as the benchmark for the quality of the fit. The difference between the single and 2-mixture Weibull distribution is apparent but small for their case study. Furthermore, they state that for their case study the 2-Weibull mixture has more symmetric errors. In Figure 5.12 it can be seen that the determined values in this study [11] indeed provide a good fit. However, it also shows a reliability of still 0.86 after 100 years which is considered unrealistic. This means that the potential application of this model is limited to the observation window of 15 years. The full data set, including all failure and censor times which are required to perform parametric estimates, could unfortunately not be retrieved to study alternatives for this data set.

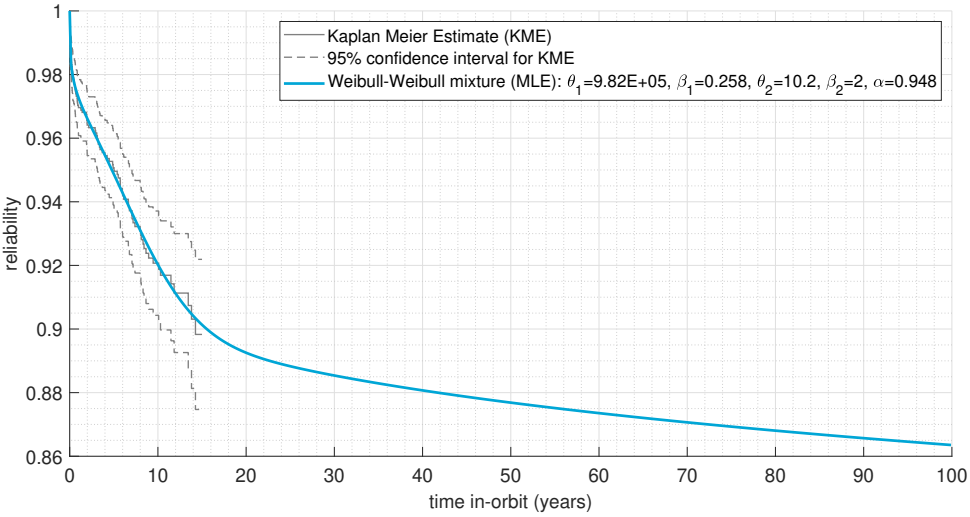
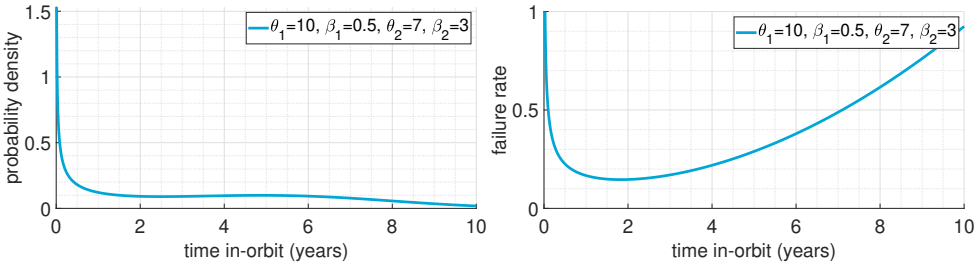


Figure 5.12: Example of Weibull-Weibull mixture based on parameter values from [11].

Reliability Products

A product of the basic reliability models would yield that all devices are subject to the risk of both immaturity failure and wear-out.



(a) probability density (b) failure rate

Figure 5.13: Example of Weibull product.

$$R(t) = \prod_{i=1}^{n} R_i(t) \tag{5.26}$$

$$F(t) = \prod_{i=1}^{n} F_i(t) = 1 - \prod_{i=1}^{n} R_i(t) \tag{5.27}$$

Using series expansion, a closed form solution is found for the probability density and failure rate.

$$\begin{aligned} f(t) = -\frac{dR(t)}{dt} = \\ -\left(\frac{R_1}{dt} \cdot R_2 \cdot \ldots \cdot R_n\right) - \left(R_1 \cdot \frac{R_2}{dt} \cdot \ldots \cdot R_n\right) - \ldots\ldots - \left(R_1 \cdot R_2 \cdot \ldots \frac{R_n}{dt}\right) = \\ \sum_{i=1}^{n} \left(\frac{f_i(t)}{R_i(t)}\right) \cdot \prod_{i=1}^{n} R_i(t) \end{aligned} \tag{5.28}$$

$$\lambda(t) = \frac{f(t)}{R(t)} = \frac{\sum_{i=1}^{n} \left(\frac{f_i(t)}{R_i(t)}\right) \cdot \prod_{i=1}^{n} R_i(t)}{\prod_{i=1}^{n} R_i(t)} = \sum_{i=1}^{n} \lambda_i(t) \tag{5.29}$$

Figure 5.13 provides an example which shows that a bathtub curve can be achieved with a Weibull product which does not have the long term declining failure rate of the Weibull mixture. Another advantage of the Weibull product is that the amount of parameters to be estimated is $2n$, as opposed to $3n - 1$ for a mixture.

A reliability product requires that the immaturity failure component leaves sufficient survivors before the peak of the probability density of the wear-out component to avoid that the latter is superfluous. This is a key difference with respect to reliability mixtures, which requires the opposite.

Extended Weibull Distribution

Instead of a compound of discrete basic reliability models, an extended Weibull distribution based on three parameters, including a nameless parameter κ, has been proposed [18] to model the bathtub curve.

$$R(t) = \exp\left\{\kappa\theta\left(1 - \exp\left[\left(\frac{t}{\theta}\right)^{\beta}\right]\right)\right\} \tag{5.30}$$

$$f(t) = \kappa\beta\left(\frac{t}{\theta}\right)^{\beta-1} \cdot \exp\left\{\left(\frac{t}{\theta}\right)^{\beta} + \kappa\theta\left(1 - \exp\left[t\left(\frac{t}{\theta}\right)^{\beta}\right]\right)\right\} \tag{5.31}$$

It has been applied by Peng et al. [19] to the same satellite data as used by Castet & Saleh [11] and shows that the maximum error and average error is lower than for the single standard Weibull model. However, the results of the extended Weibull distribution [19] are slightly less accurate than for the 2-Weibull mixture [11]. A disadvantage of the extended Weibull distribution is that it is not possible to simply derive and modify the terms responsible for the different failure classes. For these reasons, this model is not further investigated.

5

Conclusions on Compound Reliability Models

Time-divided models as well as the extended Weibull model have been discarded for further investigation. Reliability mixtures and reliability products are both considered candidates for modelling satellite reliability assuming both immaturity failure and wear-out. Reliability mixtures work well if the immaturity failure reliability component approximates zero before the wear-out does to avoid an unrealistic tail of survivors. In contrast, reliability products require that the immaturity failure component reliability is still sufficiently high before the peak of the wear-out probability density is reached to prevent that the wear-out component becomes superfluous.

To limit the number of reliability mixtures and products, basic models with a long right tail in the probability density are discarded (gamma, log-logistic and log-normal) leaving only Weibull and Gompertz as candidates. Furthermore, Gompertz is rejected for immaturity failure because it can not model a decreasing failure rate. With four basic models for immaturity failure, two for wear-out and two types of compounds (mixtures and products), a total of 16 combinations are still under investigation which will be further discussed in Section 5.5.3.

5.5. Satellite Model Estimation and Comparison

5.5.1. Estimation Methods for Parameters

There are various ways to estimate the parameters of a probability function based on a set of empirical data, such as least squares, maximum likelihood and Bayesian inference.

Least Squares Estimator

A simple mathematical approach can be performed by minimizing the sum of the squares of the residuals (SS_{res}) between the parametric reliability curve and the

Kaplan-Meier Estimate for a varying parameters vector $\boldsymbol{\theta}$. This method is called the Least Squares Estimator (LSE):

$$\mathrm{SS}_{res} = \sum_{i=1}^{n} \left(\hat{S}(t_i) - R(t_i|\boldsymbol{\theta})\right)^2 \tag{5.32}$$

$$\boldsymbol{\theta}_{LSE} = \arg\min SS_{res}(\boldsymbol{\theta}|\boldsymbol{t}). \tag{5.33}$$

Maximum Likelihood Estimation

The likelihood for a chosen set of distribution parameters for all observed events can be calculated using Equation 5.34 which is the product of probability densities of n failures at the observed times t_i and the probabilities of survival of m censored items at the censored times t_j for a chosen distribution. The distribution parameters are changed iteratively until they maximize the likelihood.

$$L(\boldsymbol{\theta}|\boldsymbol{t}) = \prod_{i=1}^{n} f(t_i|\boldsymbol{\theta}) \cdot \prod_{j=1}^{m} R(t_j|\boldsymbol{\theta}) \tag{5.34}$$

$$\boldsymbol{\theta}_{MLE} = \arg\max L(\boldsymbol{\theta}|\boldsymbol{t}) \tag{5.35}$$

For some reliability models, the probability density at $t = 0$ is zero. When failures are present at exactly $t = 0$, so will the likelihood and as a consequence the MLE will fail. The CubeSat and small satellite failure databases have a few of those entries. These satellites were either dead-on-arrival or have worked up to a few hours but failed before the first possible ground station contact. To mitigate the MLE issue and to account for limited potential operational lifetime, a bias of $+0.1$ day is applied for reported events at $t = 0$. Although this value is arbitrary, it is sufficiently high to avoid computational issues while it is insignificantly small compared to the data set observation window of many years. A second issue with some reliability models and these failures (regardless of the chosen value of the bias) is that the failure probability increases significantly when approaching $t = 0$, which can cause the MLE to put more emphasizes on these failures.

Bayesian Inference

Bayesian inference is based on the Bayes theorem [20], where a prior belief in the form of a probability distribution of the model parameters $P(\boldsymbol{\theta})$ is introduced to calculate the posterior distribution of those parameters:

$$P(\boldsymbol{\theta}|\boldsymbol{t}) = \frac{L(\boldsymbol{t}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})}{\int L(\boldsymbol{t}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})\, d\boldsymbol{\theta}} \propto L(\boldsymbol{t}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta}). \tag{5.36}$$

The integral in the denominator is taken over the complete range of possible parameter values and as such it is constant. It ensures that the posterior distribution is proper (integrates to one). It is however cumbersome to calculate this integral

and it can only be computed numerically. For most use cases, the posterior can be improper since the result will be proportional to the nominator. Likewise, the prior does not have to be proper either. Therefore, often an improper posterior is calculated which is the product of a prior and the likelihood (Equation 5.34) of the data for each set of parameter values as provided at the right side of Equation 5.36. Similar to the MLE, the Maximum-a-Posteriori (MAP) can be calculated.

$$\boldsymbol{\theta}_{MAP} = \arg \max P(\boldsymbol{\theta}|\boldsymbol{t}) \tag{5.37}$$

If a uniform distribution is taken as prior, the MAP result equals that of the MLE. The true advantage of Bayesian inference is when there is sufficient knowledge or insight to formulate an informed prior: a belief of what the distribution should look like based on previous experiments or physical (failure) models. Especially in combination with limited observations, a Bayesian method can be more powerful than MLE or LSE methods.

The mathematical tool which is used in this study is MATLAB. It has built-in LSE and MLE functions, but for Bayesian inference, a custom tool was developed. For most models, a numerical approach should be taken to calculate a representation of the posterior. With multiple parameters, the number of variations of the parameter set becomes very large and it is very computational intensive. Therefore often a Markov Chain Monte Carlo (MCMC) approach is taken using a Metropolis random walk algorithm which produces a set of output samples for which the density is proportional to the distribution [6]. The algorithm can however provide unreliable results if the proposed step-sizes in the algorithm are either too small or too large, leading to unexplored parts of the parameter space or a disproportional sample density for regions in the parameter space. In this study, it was identified that tuning the proposed distribution for more than two parameters is difficult and requires sophisticated tuning algorithms. Through experimentation it was however found that, using state-of-the-art multi-core processors of regular desktop and laptop computers, it is possible to calculate the posterior with a sufficiently fine parameter grid for models up to five parameters as used in this study. For example, a 4-parameter grid of 100 samples per parameter requires $1{\cdot}10^8$ calculations of the prior times likelihood. Even with up to 200 observations, the computation of the posterior can be performed in less than an hour on a quad-core Intel Core I7 processor running at 2.8 GHz. This method calculates the exact posterior value for each grid sample which is a very robust method. It has a second advantage in that the resulting posterior distribution array can be converted easily to a prior distribution for a different model.

Selecting an Estimator

With a very high number of failure observations, well spread over the observation time window, the results of the LSE, MLE and Bayesian methods will converge. In this case, the easy and robust LSE method can best be used. For a more limited set of data and/or a high fraction of censored items, as is the case for the CubeSat failure database, the MLE method or the Bayesian method provide a better alternative. Without an informed prior belief, the MLE method is the easiest to use.

However, if prior information on the model is available or if there is an interest in a full posterior distribution (instead of only the best estimate of the parameters), Bayesian inference should be used.

5.5.2. Comparing Model Quality

The compare the quality of the different models, several conditions and metrics can be considered which are treated in this section.

Goodness-of-fit

One way to assess the performance of a parametric model is to determine how well it fits the non-parametric model (KME). The goodness-of-fit can be determined by the coefficient of determination R^2 based on the sum of the squares of the residuals SS_{res} divided by the sum of squares w.r.t. the mean SS_{tot}. An R^2 of 1 indicates a perfect fit and a value of 0 indicates an uncorrelated fit [21].

$$R^2 = 1 - \frac{\mathrm{SS_{res}}}{\mathrm{SS_{tot}}} = 1 - \frac{\sum_{i=1}^{n} \left(\hat{S}(t) - R(t_i|\boldsymbol{\theta})\right)^2}{\sum_{i=1}^{n} \left(\hat{S}(t) - \overline{\hat{S}(t)}\right)^2} \tag{5.38}$$

Fitting can be improved by using models with more parameters. However, the added value of parameters for the model which only slightly increase the fit is limited. To prevent this, an adjusted R^2 is used [22] which includes a penalty for the number of parameters k in relation to the number of observations n.

$$R^2_{adj} = 1 - \left[\frac{(1 - R^2)(n-1)}{n-k-1}\right] \tag{5.39}$$

The R^2 can be maximized using the LSE on a specific model. The MLE includes all censored observations at the exact time of censorship explicitly in the likelihood function. LSE only indirectly takes them into account at failure times when the non-parametric result of the KME is used as input. This can explain some of the differences in R^2_{adj}, especially for an observation window with many censored items compared to failure items. However, if the R^2_{adj} of the MLE becomes significantly lower than the alternative candidates this may indicate that the MLE is biased. The vast majority of the maximum likelihood estimates of the single basic models as provided in Section 5.5.3 yield $R^2_{adj} \geq 0.95$. It is expected that a compound model has a better goodness of fit, so $R^2_{adj} \geq 0.95$ is used as acceptance criterion for selecting an appropriate compound model.

Mode of Wear-out Component

The mode of the distribution is the time where the probability density peaks. For immaturity failure this is at or near $t = 0$. For the wear-out the mode should not occur too early as this would interfere significantly with immaturity failure. It should also not occur unrealistically late. However, it is plausible that it lies outside the observation time window of the CubeSat failure database as there are still many satellites operational at the end of this window. For this study a range between 1 and 25 years is chosen as acceptance criterion for the mode.

Long-term Reliability

A good compound model assures that the vast majority of the satellites have worn out in a reasonable time, to avoid that simulations using the best estimate of the model contain too many extreme outliers. The acceptance criterion used in this study is that the reliability $R_{t=50y} \leq 0.1\%$.

Wear-out Shape Parameter Boundary

For both the CubeSat and small satellite failure databases there are many failures in the first year, which causes the MLE to naturally converge towards immaturity failure values for both components. To ensure that the second component of the compound model addresses wear-out, a boundary condition is used in the MLE for Weibull ($\beta_2 \geq 2$) and Gompertz ($\eta_2 \leq 0.1$). Ideally the MLE does not converge to this boundary condition but finds a true (local) maximum.

Akaike Information Criterion

The best ranking criterion for models using MLE is the likelihood L. However, similar to the goodness-of-fit, likelihood can be improved by adding parameters with the risk that they provide little extra information. To deal with this issue, the Akaike Information Criterion (AIC) [23] can be used where the number of parameters k is included as a penalty. The AIC value has no meaning in absolute sense, but the AIC of different models can be compared in relative sense where a lower value is better:

$$\text{AIC} = 2k - 2 \cdot \ln(L). \tag{5.40}$$

Another method is the Bayesian Information Criterion (BIC) which which also includes the sample size n:

$$\text{BIC} = k \cdot ln(n) - 2 \cdot \ln(L). \tag{5.41}$$

According to Aho [23], BIC is used for selecting the correct model out of completely specified models while AIC is used to determine the best models which lack complete specification. The latter is the case in this study, so AIC is chosen as criterion. Taking the best model as reference, all models which have an AIC value of $+6$ or higher are rejected as this yields a likelihood ratio of $< 5\%$ compared to the best model in terms of AIC based on an equal number of parameters. Comparing mixture and product compounds with two components of the chosen distributions yields five and four independent parameters respectively.

The model with the lowest AIC, which meets the previously described acceptance criteria, is chosen as the preferred model. The focus of this study is the CubeSat failure database. However, the small satellite failure database is also analyzed.

5.5.3. Results & Analysis using MLE

The results of the chosen set of compound models (Section 5.4.2), using the maximum-likelihood-estimator, are shown in Figures 5.14, 5.15, 5.16 and 5.17 for CubeSats. Figures 5.18, 5.19, 5.20 and 5.21 show the estimates for small satellites.

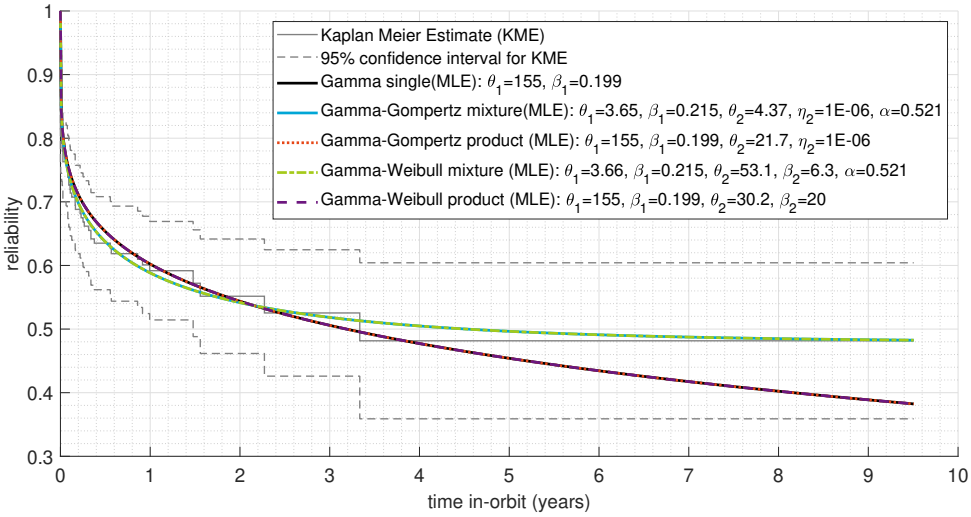


Figure 5.14: MLE estimates of CubeSats using the Gamma distribution for immaturity failure.

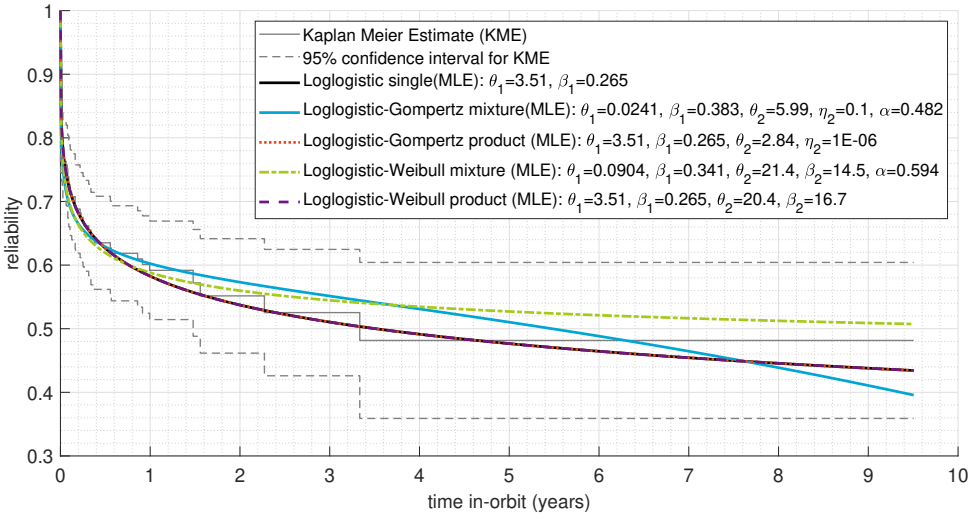


Figure 5.15: MLE estimates of CubeSats using the Loglogistic distribution for immaturity failure.

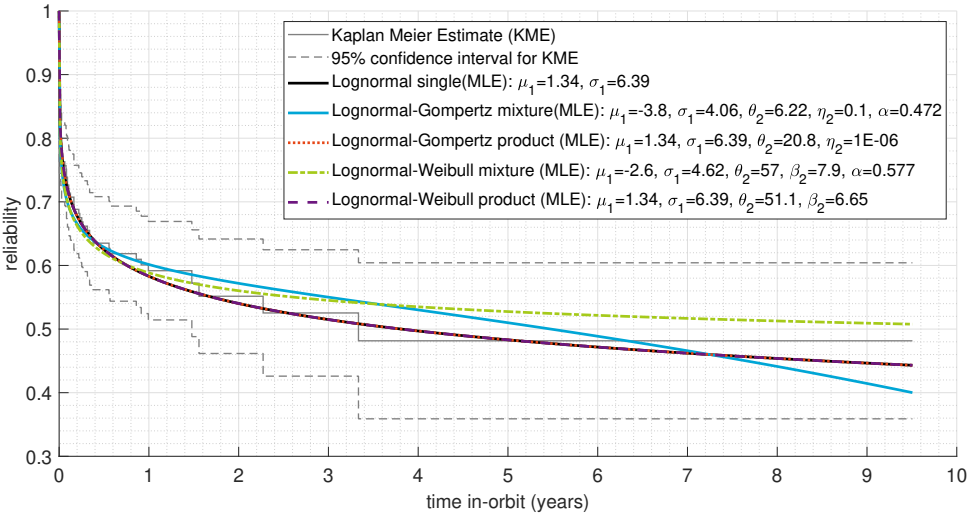


Figure 5.16: MLE estimates of CubeSats using the Lognormal distribution for immaturity failure.

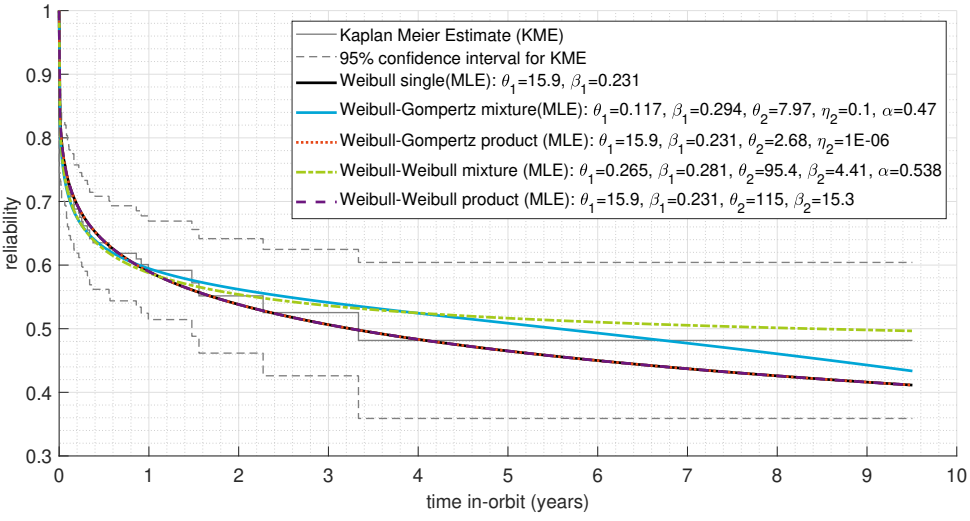


Figure 5.17: MLE estimates of CubeSats using the Weibull distribution for immaturity failure.

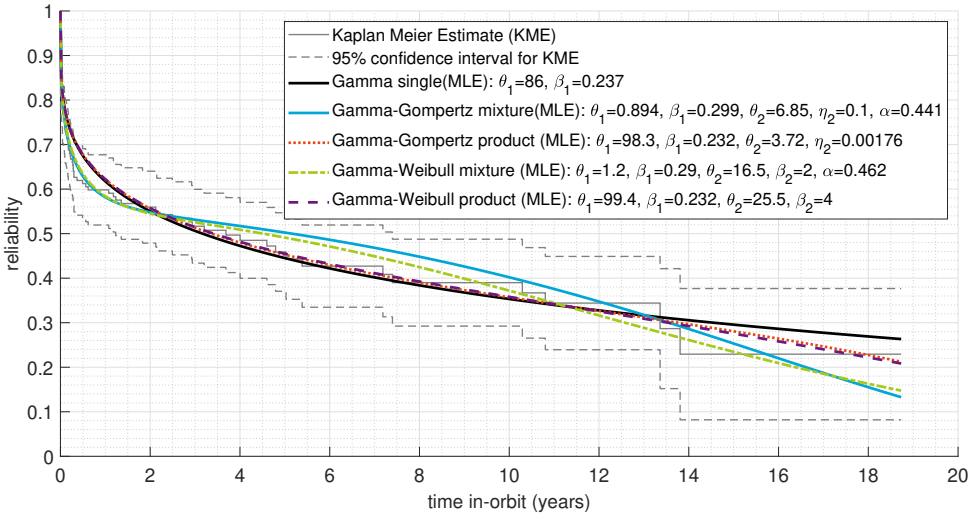


Figure 5.18: MLE estimates of small satellites using the Gamma distribution for immaturity failure.

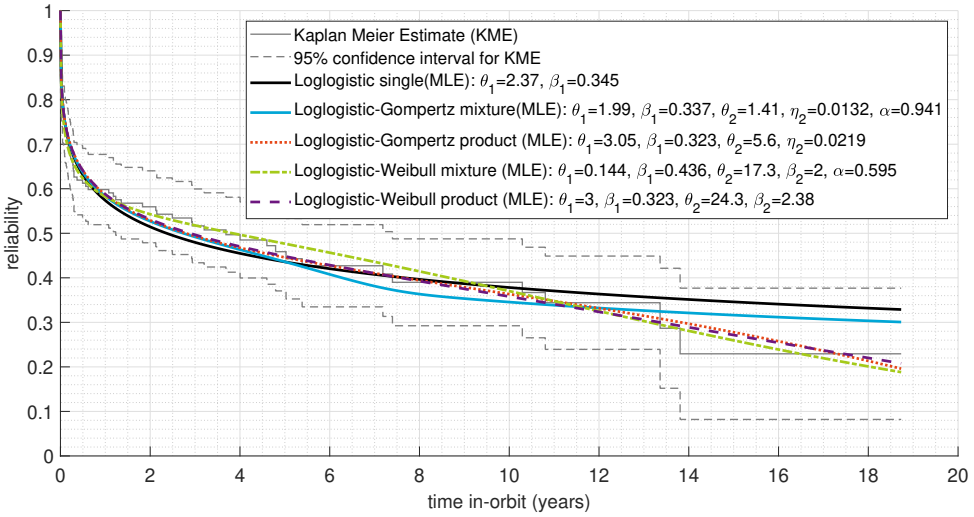


Figure 5.19: MLE estimates of small satellites using the Loglogistic distribution for immaturity failure.

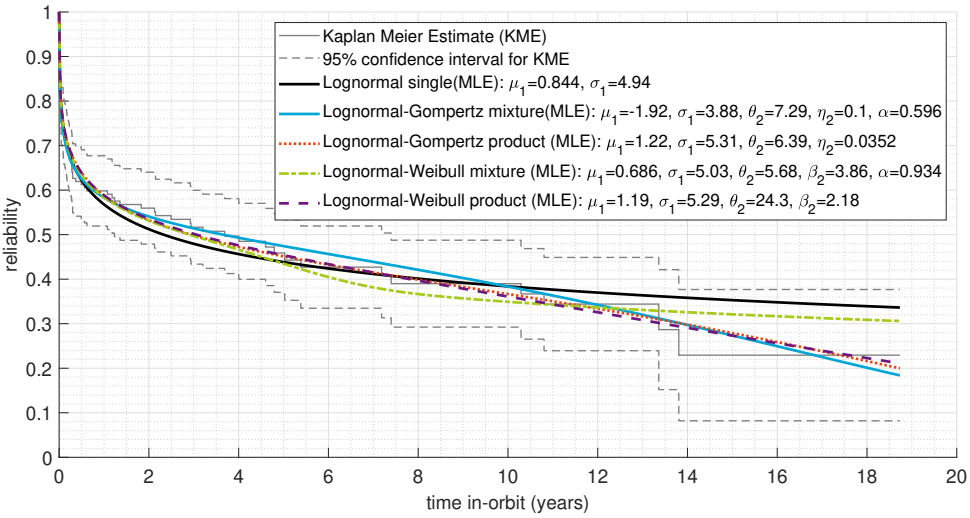


Figure 5.20: MLE estimates of small satellites using the Lognormal distribution for immaturity failure.

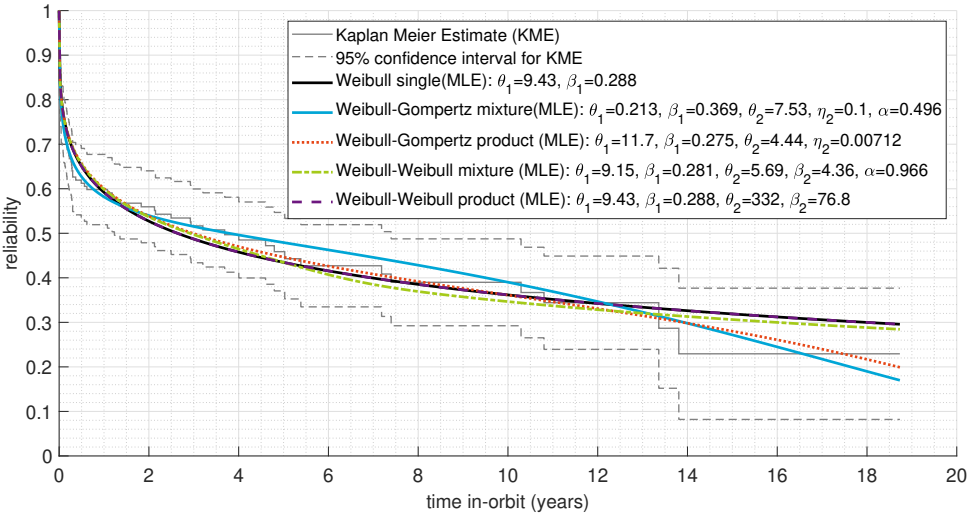


Figure 5.21: MLE estimates of small satellites using the Weibull distribution for immaturity failure.

The Akaike Information Criterion AIC, the goodness-of-fit R^2_{adj}, the mode of the wear-out component $Mode_2$, the reliability at 50 years $R_{t=50y}$ and a 'wear-out shape factor at boundary condition' check are provided in Table 5.1. The list is ordered on AIC value, with the lowest (i.e. best) at the top. If the acceptance criteria for the other values are violated, the specific value is colored red.

The best AIC values for both databases are provided by the Lognormal distribution for immaturity failure. Another general conclusion which can be made is that MLE results for mixtures have a tendency to converge to the boundary condition for the wear-out shape factor and they yield unrealistically high reliability on the long term ($R_{t=50y} > 1\%$). In this respect, reliability products are performing significantly better. For the CubeSat failure database, which is the main focus of this study, unfortunately all models violate at least one of the acceptance criteria. The single Lognormal model even has the best AIC and also one of the highest R^2_{adj}. This indicates that the CubeSat failure database does not contain sufficient information (relatively low number of failures beyond the first year) to provide convincing results of the wear-out parameter values. For the small satellite failure database however, the Lognormal-Gompertz product meets all acceptance criteria and comes very close to the highest AIC. It combines a good fit for immaturity failure with the property of a relatively short survivor tail for the wear-out compared to other models.

5.5.4. Results & Analysis using Bayesian Inference

As there is insufficient data on CubeSat failures to provide appropriate estimates using the MLE method, another approach is investigated. Using Bayesian inference, the small satellite failure database could be used to provide a prior distribution for CubeSats based on the fact that CubeSats are a subset of the class of small satellites. The model parameters for CubeSats should lie within a range of values which have a reasonable likelihood ratio compared to the maximum likelihood estimate of the small satellites. First, the posterior distribution of the small satellite database is calculated using a uniform prior. With this uninformed prior, the maximimum-a-posteriori (MAP) parameters should match the maximum likelihood estimate (MLE). It also provides a four-dimensional posterior (because of having four parameters) which can be converted to a prior for the CubeSat failure database. The normalized marginal posterior distribution, which is the sum for one parameter over the other parameter values normalized to its maximum, are provided in Figure 5.22. It should be noted that the MAP is found in a four-dimensional posterior, and may therefore differ from the maximum of the marginal distribution.

It can be seen that the MAP values in Figure 5.22 closely match the MLE values in Figure 5.20 as expected, with very minor differences which are fully explained by the grid step size for the Bayesian inference.

Using the posterior distribution of the small satellite failure database directly as prior for the CubeSat failure database would effectively yield the same result as combining both databases into one. There are only 20 CubeSats in the small satellite database of 152 satellites, significantly less than the 178 in the CubeSat database. The posterior should therefore first be weakened to be able to act as

Table 5.1: Results of the MLE estimates.

CubeSat failure database

Model	AIC	R^2_{adj}	$Mode_2$	$R_{t=50y}$	Boundary
Lognormal single	-95.4	0.959		34.35%	
Lognormal-Gompertz mixture	-93.7	0.904	14.3	1.36%	yes
Lognormal-Weibull mixture	-92.3	0.914	56.0	34.17%	no
Lognormal-Gompertz product	-91.4	0.958	287.1	34.35%	no
Lognormal-Weibull product	-91.4	0.958	49.8	14.41%	no
Loglogistic single	-89.1	0.962		33.08%	
Loglogistic-Gompertz mixture	-87.6	0.901	13.8	2.45%	yes
Loglogistic-Weibull mixture	-86.1	0.919	21.3	6.19%	no
Weibull single	-86.1	0.953		27.16%	
Loglogistic-Weibull product	-85.1	0.961	20.4	0.00%	no
Loglogistic-Gompertz product	-85.1	0.961	39.2	0.00%	no
Weibull-Gompertz mixture	-83.9	0.944	18.4	0.12%	yes
Weibull-Weibull mixture	-83.3	0.949	90.0	44.25%	no
Gamma single	-82.3	0.913		17.41%	
Weibull-Weibull product	-82.1	0.952	114.5	27.16%	no
Weibull-Gompertz product	-82.1	0.952	37.0	0.00%	no
Gamma-Weibull product	-78.3	0.911	30.2	0.00%	no
Gamma-Gompertz product	-78.3	0.911	299.6	17.41%	no
Gamma-Gompertz mixture	-78.1	0.955	60.3	43.60%	no
Gamma-Weibull mixture	-78.1	0.955	51.7	24.19%	no

Small satellite failure database

Model	AIC	R^2_{adj}	$Mode_2$	$R_{t=50y}$	Boundary
Lognormal-Gompertz mixture	171.5	0.985	16.8	3.97%	yes
Lognormal-Weibull product	171.6	0.984	18.4	0.25%	no
Lognormal-Gompertz product	172.1	0.982	21.4	0.00%	no
Lognormal single	172.6	0.949		26.72%	
Loglogistic-Weibull mixture	174.1	0.987	12.2	4.32%	yes
Loglogistic-Weibull product	176.1	0.981	19.3	0.11%	no
Lognormal-Weibull mixture	176.1	0.970	5.3	24.34%	no
Loglogistic single	176.3	0.951		25.88%	
Weibull-Gompertz mixture	176.4	0.982	17.3	0.03%	yes
Loglogistic-Gompertz product	176.5	0.979	21.4	0.00%	no
Weibull single	177.4	0.959		19.87%	
Weibull-Gompertz product	179.8	0.972	22.0	0.00%	no
Gamma single	180.3	0.954		12.50%	
Loglogistic-Gompertz mixture	180.6	0.967	6.1	23.74%	no
Weibull-Weibull product	181.4	0.958	331.6	19.87%	no
Gamma-Weibull mixture	181.7	0.972	11.6	0.01%	yes
Gamma-Gompertz mixture	182.6	0.963	15.8	0.00%	yes
Weibull-Weibull mixture	182.8	0.963	5.4	19.28%	no
Gamma-Weibull product	183.8	0.955	23.7	0.00%	no
Gamma-Gompertz product	184.0	0.955	23.6	0.00%	no

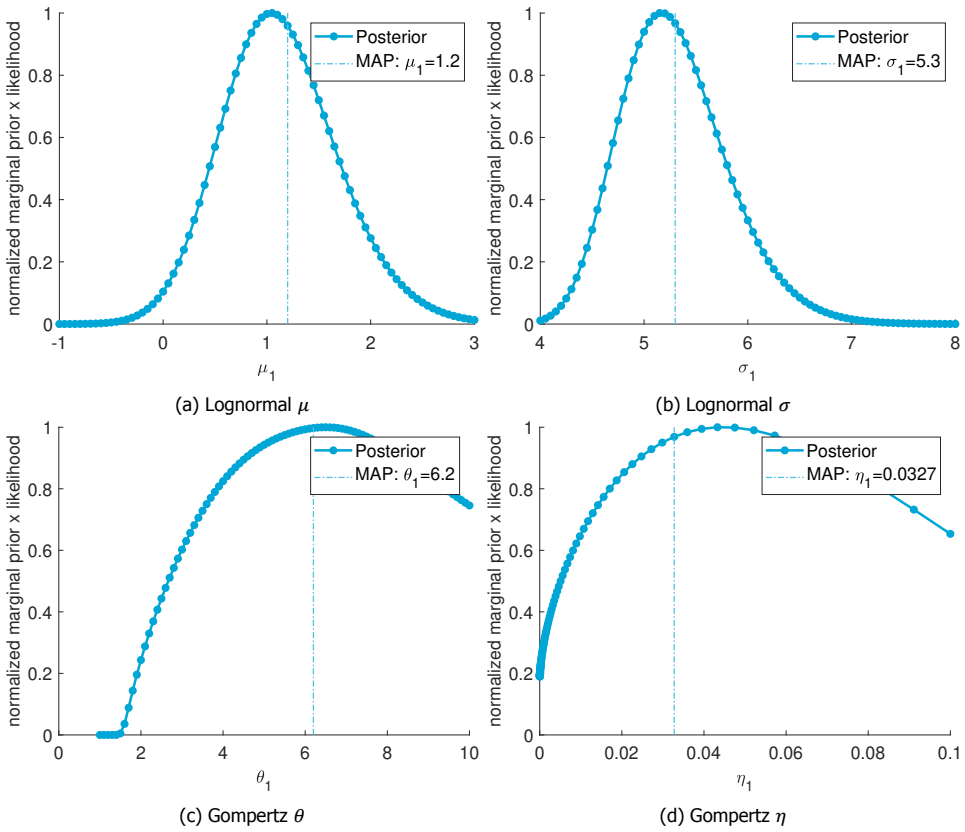


Figure 5.22: Marginalized posterior distributions for small satellite failure database.

prior. There is no common method or clear rules as Bayesian inference relies on the additional knowledge and/or logical reasoning to define the prior. The posterior is proportional to the likelihood times the prior as explained in Section 5.36 and the likelihood is the product of probabilities of failure or survival. Given this product relationship, a weakening of each value of the posterior using an weight exponent w between 0 and 1 would be a natural choice to achieve a new prior:

$$p_{prior}(\boldsymbol{\theta}) = p_{post}(\boldsymbol{\theta})^w. \tag{5.42}$$

Next, w needs to be determined to use the posterior of the small satellite database as prior for the Bayesian inference of CubeSats failures. Using Equation 5.42 and a weight of 20/152, the posterior of the small satellite database is weakened to the order of a likelihood of 20 satellites (provided that the prior for the small satellite data set was uniform). Figure 5.23 shows the translation (normalized to the MAP) for using the selected weight as well as ±25% of this weight which is

used to analyse the sensitivity of the result to change of the selected weight:

Table 5.2 and Figure 5.25 provide the results in comparison with the MLE for CubeSats and the MAP of the small satellites. The estimate using $w = 20/152$ meets all acceptance criteria. Its lognormal parameters come very close to the MLE, which is expected due the relative high number of early failures. The wear-out parameters are now realistic w.r.t. the MLE, while the AIC comes very close. The overall distribution does not show a high sensitivity to the chosen translation weight when changed by ±25%. Figure 5.24 shows the marginal distributions of the CubeSat posterior, the used prior and the posterior of the small satellites which the prior is based upon.

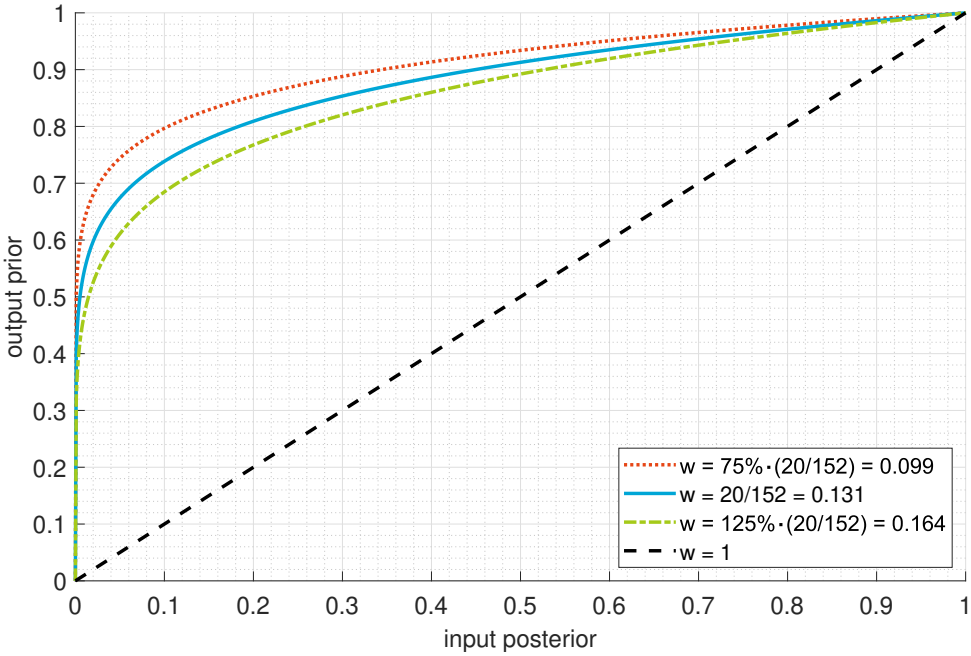
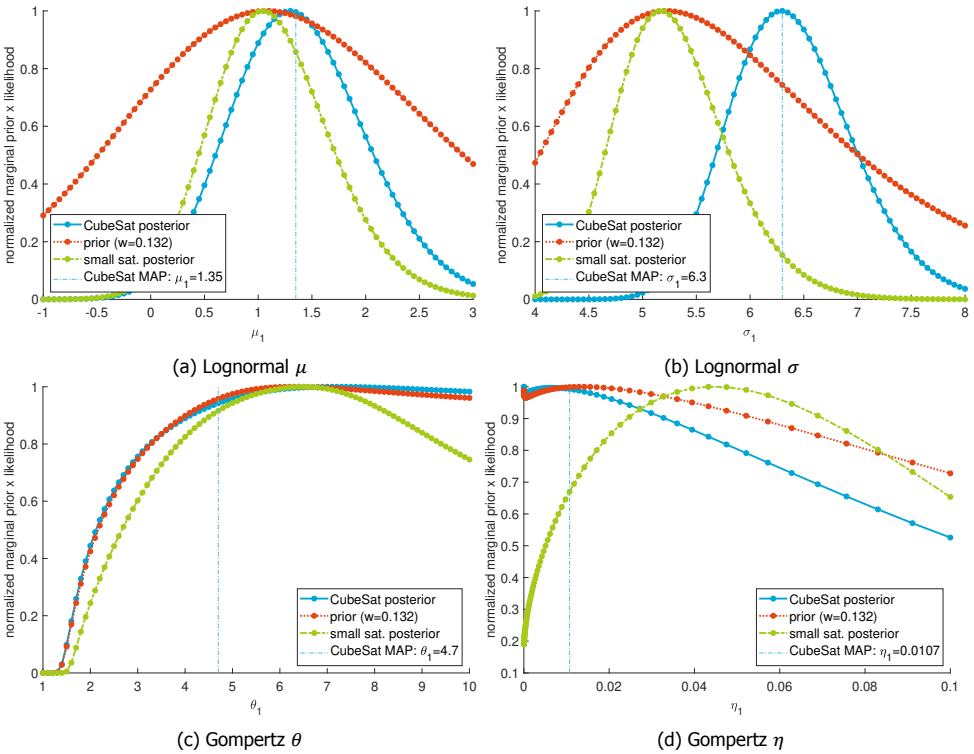


Figure 5.23: Posterior-to-prior translation, normalized to MAP.

Table 5.2: Results of the MAP estimates for Cubesat failure database

Estimate	AIC	R^2_{adj}	$Mode_2$	$R_{t=50y}$	μ_1	σ_1	θ_2	η_2
MLE	-91.4	0.958	287.1	34.35%	1.30	6.40	20.8	$1.00 \cdot 10^{-6}$
75% of w	-91.2	0.950	21.1	0.00%	1.30	6.30	4.0	$5.10 \cdot 10^{-3}$
$w = 20/152$	-91.1	0.951	21.3	0.00%	1.35	6.30	4.7	$1.07 \cdot 10^{-2}$
125% of w	-91.1	0.946	21.3	0.00%	1.30	6.25	4.9	$1.29 \cdot 10^{-2}$
small sat.	-85.0	0.889	21.2	0.00%	1.20	5.30	6.2	$3.27 \cdot 10^{-2}$

In conclusion, the best posterior distribution for the CubeSat failures is found through Bayesian inference on the CubeSat failure database using the small satellite

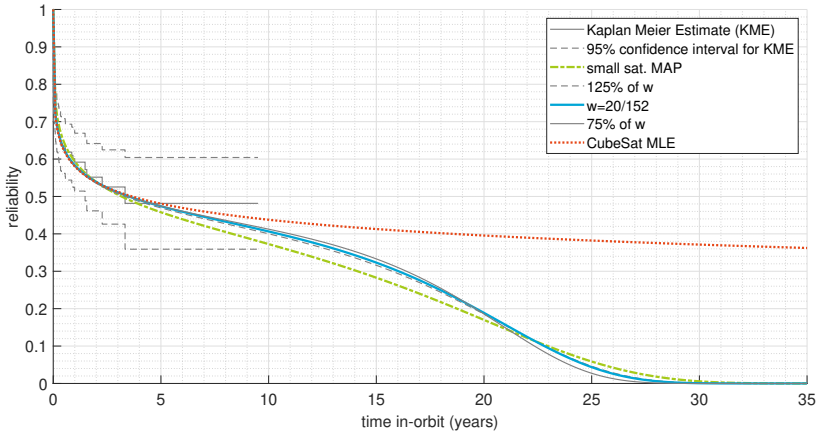


(a) Lognormal μ (b) Lognormal σ

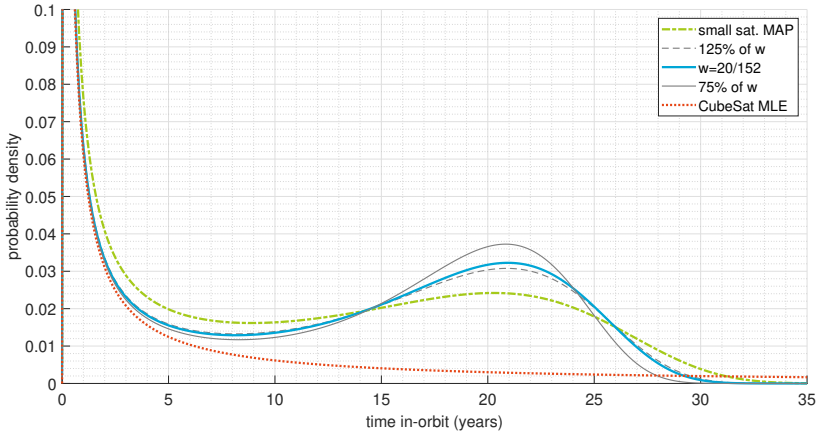
(c) Gompertz θ (d) Gompertz η

Figure 5.24: Marginalized prior and posterior distributions, normalized to their MAP.

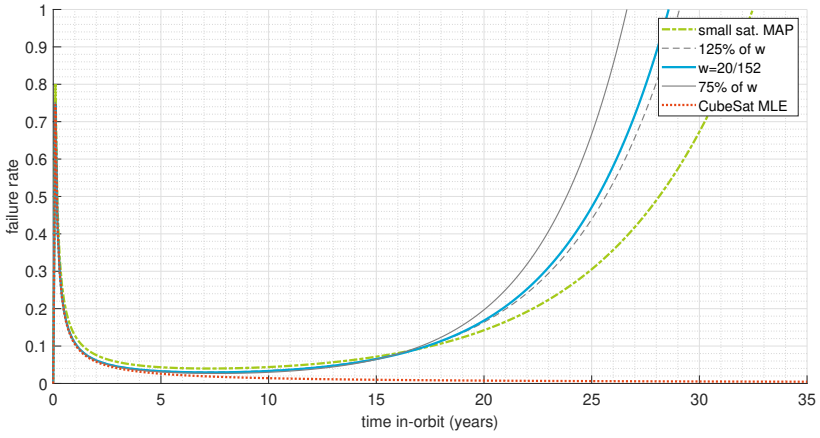
failure database posterior as input which is translated to a prior using Equation 5.42 with $w = 20/152$. The resulting maximum-a-posteriori estimate is a Lognormal-Gompertz product with parameters $\mu_1 = 1.35$, $\sigma_1 = 6.30$, $\theta_2 = 4.7$ and $\eta_2 = 0.0107$.



(a) reliability



(b) probability density



(c) failure rate

Figure 5.25: CubeSat MAP distributions for different priors and the MLE for reference.

5

5.6. Subsystem Model Estimation

Having determined satellite reliability, the next step is to determine subsystem reliability. Equation 5.43 provides the general relation between the system reliability R_{sys} and its subsystems reliability $R_{ss,i}$ for all n subsystems:

$$R_{sys}(t) = \prod_{i=1}^{n} R_{ss,i}(t). \tag{5.43}$$

A first subsystem reliability estimate can be made by assuming all $R_{ss,i}$ are identical. This estimate can be used as prior for Bayesian inferences on specific subsystem failure data. The CubeSat failure database contains information on the determined or suspected subsystem that led to a satellite failure: Attitude Determination and Control (ADCS), Command and Data Handling Subsystems (CDHS), Communication Subsystem (COMMS), Structure & Deployment Mechanisms (STS & DepS), Electrical Power Subsystem (EPS) and Payload (P/L). The Thermal Control Subsystem (TCS) is missing in the database because this subsystem is typically passive in CubeSats or embedded in other subsystems. For example, a battery heater would be considered part of the EPS. The electrical interfaces between subsystems are also allocated to the major subsystems (e.g. data interfaces to CDHS, power distribution to EPS). For a study on the effect of redundancy of subsystems, the allocation of failures to these subsystems is not ideal as redundancy is typically applied to physical units and their interfaces. It is expected that more advanced CubeSats may have more physical units and/or more sophisticated units. For example, an advanced ADCS may comprise an additional board with reaction wheels. If n would represent physical units, its value would differ per satellite. On the other hand, there is a correlation between the sophistication of CubeSats and the experience of its developers as functionality is often added in follow-up satellite projects. Because of the lack of insight of failures related to all these aspects, the potential analysis is limited to the breakdown in aforementioned subsystems. For the research goal of this chapter, investigating the impact of subsystem redundancy for CubeSats in general, the limited breakdown is considered to be acceptable. When assessing the reliability of a specific CubeSat, the estimates in this study should be complemented with insights in the complexity of the design, team experience and intensity and results in testing. An example of CubeSat specific reliability estimation and growth is provided by Langer [24].

Besides the breakdown in six subsystems, the database also contains a category 'unknown' for satellite failures in which the fatal subsystem is unknown. With 23 out of 71 failures classified as unknown, this is the largest group. Censoring the items would lead to a considerable overestimation of the reliability, so removal of these entries from the database for subsystem analysis is therefore considered to be the best solution. A final check is required if the product of subsystem reliability estimates approximates the general CubeSat reliability estimate.

Equation 5.43 can only be used for non-redundant subsystems or the aggregated reliability of redundant subsystems. The database does not contain any information on subsystem redundancy. According to the CubeSat survey on data

busses, approximately 10% of the implemented data busses are redundant (see Section 2.3.1). While this does not allow to draw any conclusion about redundancy in other subsystems, it may be used as an indication that full redundancy is not widely implemented yet. For none of the reported satellite failures, a dual failure of a redundant subsystem was mentioned as cause. Furthermore, the impact of redundancy of subsystems on satellite lifetime extension is expected to be highest for the population subject to wear-out, for which the database provides little information as discussed in Section 5.5. For immaturity failures, it is assumed that the vast majority of CubeSat failures are due to single subsystem failures or common mode failures. Equation 5.43 is therefore used to estimate the reliability model parameters of non-redundant subsystems.

The posterior for satellites is converted to a prior for subsystems and Bayesian inference is subsequently applied using the specific subsystem failure data. The posterior for the satellite reliability parameters is translated into an equivalent posterior for subsystems based on identical distributions. In this case Equation 5.43 translates into Equation 5.44 and, using Equation 5.5, into Equation 5.45.

$$R_{sat}(t) = R_{ss}(t)^n \tag{5.44}$$

$$f_{sat}(t) = -\frac{dR_{sat}(t)}{dt} = -n \cdot \frac{dR_{ss}(t)}{dt} \cdot R_{ss}(t)^{n-1} = n \cdot f_{ss}(t) \cdot R_{ss}(t)^{n-1} \tag{5.45}$$

For immaturity failure $imm.$ and the wear-out $w.o.$, subsystem reliability can be split by Equation 5.46. This means that the posterior from the results in Section 5.5.4 can be used as-is by calculating the subsystem parameters associated with the satellite parameters for immaturity and wear-out separately.

$$R_{sat}(t) = [R_{ss,imm.}(t) \cdot R_{ss,w.o.}(t)]^n = R_{ss,imm.}(t)^n \cdot R_{ss,w.o.}(t)^n \tag{5.46}$$

For wear-out, Equation 5.47 holds when $\theta_{sat} = \theta_{ss}$ and $\eta_{sat} = n \cdot \eta_{ss}$.

$$R_{sat,w.o.}(t) = \exp\left[-\eta_{sat}\left(e^{\left(\frac{t}{\theta sat}\right)} - 1\right)\right] = \exp\left[-\eta_{ss}\left(e^{\left(\frac{t}{\theta_{ss}}\right)} - 1\right)\right]^n \tag{5.47}$$

For immaturity failure, the integral of the Lognormal reliability function cannot be solved in closed form. Instead, the subsystem parameters can be calculated numerically by a discrete representation of the curve for R_{sat} for each set of $(\mu_{sat}, \sigma_{sat})$ in the parameter grid of the satellite posterior. Subsequently, the least squares estimator is used to find the parameters (μ_{ss}, σ_{ss}) for n subsystems for each grid location. This method has been performed using 1000 data points on a logarithmic scale between 0.001 and 100 years. The resulting grid values for μ_{ss} and σ_{ss}

values are dependent on both μ_{sat} and σ_{sat}, so subsequent Bayesian inference with subsystem data should be based on the original satellite parameter grid which is then converted to subsystem estimates point-by-point. With this approach, the posterior can be calculated and subsystem MAPs can be found for each point, but the new posterior distribution cannot be marginalized for μ_{ss} and σ_{ss}. The R^2_{adj} values for each new distribution based on the reliability product of $n = 6$ subsystems with respect to the to the original reliability of satellites ranges from 0.9991 to 0.9996. This is considered to be a near perfect fit. Figure 5.27 shows that the difference between the satellite reliability MAP and the product of the approximated subsystem reliability parameters is indeed small. Using this approach the satellite MAP corresponds to subsystem parameters of $\mu_1 = 13.2$, $\sigma_1 = 9.59$, $\theta_2 = 4.7$ and $\eta_2 = 0.0018$.

The satellite posterior should be weakened to act as prior, as explained in Section 5.5.4. In this case a weight of $w = 1/n = 1/6$ is applied for Equation 5.42. Using this approach, the failure data will dominate over the prior when there are relative more failures allocated to a subsystem than one-sixth of the satellite failures. If there are less failures for a subsystem, the prior will dominate. Again, the results with ±25% of this weight are also calculated to determine the sensitivity of the results with respect to the weight.

The limited number of failures for each subsystem limits the confidence of the Kaplan Meier Estimate. Secondly, the large proportion of failures allocated to an unknown subsystem (32%) means that the KME is too optimistic for some subsystems and the lower confidence bound is too high for all. Worst case, if all unknown failures would be allocated to a specific subsystem, the reliability could be 0.32 lower at the end of the observation window. For these reasons, the subsystem KME is not a good reference and the goodness-of-fit loses its meaning and is therefore not provided. The remaining results are presented in Table 5.3, which provides the Bayesian MAP estimate using $w = 1/6 \pm 25\%$ as well as the MAP of the prior (the parameters for all subsystem distributions identical) and the MLE of the subsystem failure data.

From Table 5.3 it can be seen that MLE provides unrealistically high values for the wear-out mode $Mode_2$ and the reliability after 50 years $R_{t=50y}$. For the Bayesian estimates using $w = 1/6$ this is all near zero which is more plausible. The results are not significantly sensitive to varying of w while the AIC value is closer to the MLE than to the prior MAP for subsystems with more failure data. Figure 5.26 provides the MAP estimates using $w = 1/6$ as well as the KME.

Because the database only contains two failures allocated to ADCS, STS & DepS and the payload, the reliability curves are similar and drops below the KME lower confidence bound. This lower bound is however unrealistic because of the relatively high number of un-allocated failures as well as the constant confidence bound in absence of failures. For the subsystems with more allocated failures, EPS (20), CDHS (12) and COMMS (10), the curves show more variation and approximate the KME better. Overall, the results can be considered plausible. Using these new estimates, the subsystem product reliability can be calculated and compared to the satellite reliability estimate. This is shown in Figure 5.27.

Table 5.3: Results of the MAP estimates for CubeSat subsystem failures.

S/S	Estimate	AIC	$Mode_2$	$R_{t=50y}$	μ_1	σ_1	θ_2	η_2
	S/S MLE	8.2	563.4	97.56%	42.24	19.45	40.8	$1.00 \cdot 10^{-6}$
	75% of w	15.7	23.7	0.00%	15.07	9.82	2.4	$5.21 \cdot 10^{-5}$
ADCS	$w = 1/6$	15.9	24.2	0.00%	15.35	10.05	2.6	$9.10 \cdot 10^{-5}$
	125% of w	15.9	24.8	0.00%	15.45	10.12	2.9	$1.92 \cdot 10^{-4}$
	prior MAP	20.6	29.7	0.00%	13.21	9.59	4.7	$1.79 \cdot 10^{-3}$
	S/S MLE	72.3	26.7	0.05%	9.43	6.68	11.6	$1.00 \cdot 10^{-1}$
	75% of w	73.8	31.5	0.00%	11.21	8.16	7.7	$1.67 \cdot 10^{-2}$
CDHS	$w = 1/6$	74.0	33.2	0.03%	11.53	8.39	8.1	$1.67 \cdot 10^{-2}$
	125% of w	74.2	34.4	0.14%	11.76	8.53	8.4	$1.67 \cdot 10^{-2}$
	prior MAP	76.0	29.7	0.00%	13.21	9.59	4.7	$1.79 \cdot 10^{-3}$
	S/S MLE	28.9	187.5	85.40%	14.66	10.20	13.6	$1.00 \cdot 10^{-6}$
	75% of w	29.0	24.2	0.00%	13.86	9.85	2.5	$6.27 \cdot 10^{-5}$
COMMS	$w = 1/6$	29.0	24.4	0.00%	13.69	9.79	2.6	$8.30 \cdot 10^{-5}$
	125% of w	29.0	24.6	0.00%	13.56	9.72	2.7	$1.10 \cdot 10^{-4}$
	prior MAP	29.2	29.7	0.00%	13.21	9.59	4.7	$1.79 \cdot 10^{-3}$
	S/S MLE	27.3	2766.6	91.58%	10.98	5.13	200.3	$1.00 \cdot 10^{-6}$
STS	75% of w	34.3	23.9	0.00%	13.85	8.83	2.4	$4.75 \cdot 10^{-5}$
&	$w = 1/6$	34.8	24.6	0.00%	14.32	9.21	2.7	$1.10 \cdot 10^{-4}$
DepS	125% of w	35.1	25.1	0.00%	14.60	9.43	2.9	$1.75 \cdot 10^{-4}$
	prior MAP	40.7	29.7	0.00%	13.21	9.59	4.7	$1.79 \cdot 10^{-3}$
	S/S MLE	63.0	187.0	63.77%	7.53	7.14	80.3	$9.76 \cdot 10^{-2}$
	75% of w	64.1	25.6	0.00%	9.40	7.97	2.6	$5.21 \cdot 10^{-5}$
EPS	$w = 1/6$	64.6	26.4	0.00%	9.85	8.18	2.9	$1.10 \cdot 10^{-4}$
	125% of w	65.0	27.1	0.00%	10.17	8.32	3.2	$2.10 \cdot 10^{-4}$
	prior MAP	69.5	29.7	0.00%	13.21	9.59	4.7	$1.79 \cdot 10^{-3}$
	S/S MLE	27.3	2766.6	91.58%	10.98	5.13	200.3	$1.00 \cdot 10^{-6}$
	75% of w	34.3	23.9	0.00%	13.85	8.83	2.4	$4.75 \cdot 10^{-5}$
P/L	$w = 1/6$	34.8	24.6	0.00%	14.32	9.21	2.7	$1.10 \cdot 10^{-4}$
	125% of w	35.1	25.1	0.00%	14.60	9.43	2.9	$1.75 \cdot 10^{-4}$
	prior MAP	40.7	29.7	0.00%	13.21	9.59	4.7	$1.79 \cdot 10^{-3}$

The goodness-of-fit of the product of identical subsystem reliability compared to its original satellite estimate is $R_2 = 0.997$ for the first 10 years and $R_2 = 0.998$ for the first 30 years. The goodness-of-fit of the product of the individual subsystem reliability estimates compared to the satellite estimate is $R_2 = 0.958$ for 10 years and $R_2 = 0.984$ for 30 years. While the difference in the curves can clearly be seen in Figure 5.27, this difference is considered acceptable given the limitations in the subsystem failure data. The LSE of the individual subsystem product is $\mu_1 = 1.51$, $\sigma_1 = 5.50$, $\theta_2 = 3.03$ and $\eta_2 = 0.0014$ which is a perfect fit ($R^2 = 1.00$).

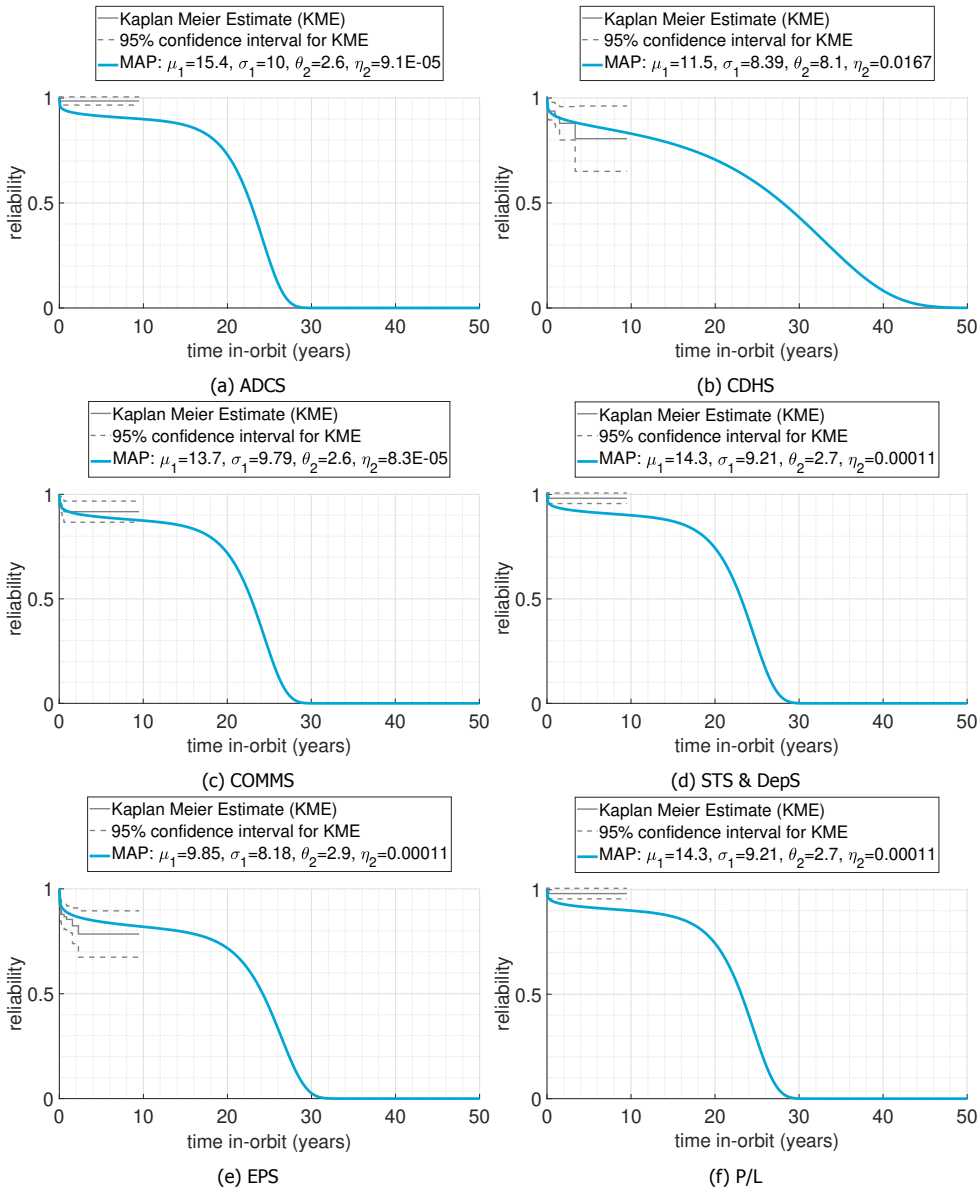


Figure 5.26: Reliability MAP estimates using $w = 1/6$.

For the subsequent steps in modelling, the subsystem MAP estimates from Table 5.3 with $w = 1/6$ can be used as best estimate. Samples from the posteriors can be used to also include the parameter uncertainty in the simulation. These samples can be obtained by applying the following procedure for each subsystem:

1. Draw a random parameter set $(\mu, \sigma, \beta, \eta)$ from the parameter grid.
2. Calculate the likelihood ratio λ for the sample with respect the MAP.
3. Draw a random number p from a uniform distribution [0,1].
4. Select the parameter set if $\lambda > p$.
5. Repeat steps 1-4 until a defined number of sets have been selected.

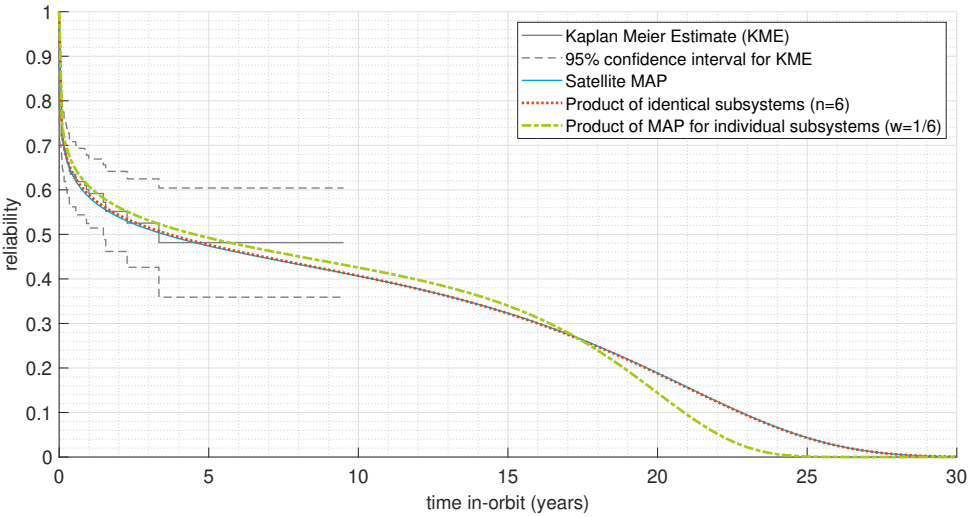


Figure 5.27: Comparison of product of CubeSat subsystems reliability estimates with satellite estimate.

5.7. Satellite Reliability Model and Scenarios

The reliability models for subsystems have been established in Section 5.6. The next step is to create a reliability simulation for the satellite, using this data as input, to answer the main question for this chapter "What leads to higher CubeSat reliability over its mission life time: full subsystem redundancy or improved testing?".

The subsystem redundancy concept is explained first. This is followed by investigating the failure dependency of a redundant subsystem. Subsequently, a model is established for CubeSats with non-redundant subsystems which accounts for the effect of more extensive testing. Finally, the results are compared and the research question is answered.

5.7.1. Subsystem Redundancy Concept

There are many ways how redundancy can be implemented. Redundancy can be implemented at subsystem and component level, using identical or different units and operated in cold, warm or hot configuration. The difference between cold,

warm or hot redundancy relates to the operational mode in which the secondary unit is kept when the primary still functions nominally. With cold redundancy, the secondary unit remains off until it is activated. With hot redundancy, the secondary unit is fully activated and updated on the states of the primary unit, enabling seamless take-over when needed. Warm redundancy means that the secondary unit is in standby mode and only updated on critical aspects. Due to the limited available electrical power on CubeSats and PocketQubes, cold redundancy is assumed for the reliability study. This also has as advantage that the redundant systems are not subject to operational wear. Other than that, the choice mostly impacts the way of operations and not necessarily the reliability of the satellite in terms of its mission objective.

For redundancy of subsystems, it is assumed that identical units are chosen. The reason is that identical units will provide exactly the same performance and are operated in the same way. It is also possible to choose different units, from different developers but compliant to the full mission needs, to mitigate failure dependencies. However, this is very complex and costly to implement and considered to be outside the scope of CubeSats. A third strategy is to implement a different redundant unit which is not compliant to the full mission needs, but may be able to save key mission objectives. In this study, only the satellite reliability in terms of the full mission is considered.

When redundancy is applied to the full subsystem unit(s) and its components (e.g. sun sensors), it is possible to use cross-strapping between them using redundant electrical interfaces. This is for example applied on the Flying Laptop Platform (FLP) satellite [25] and the Mars Reconnaissance Orbiter [26]. This is a small satellite of 117 kg and a medium sized satellite of 2180 kg respectively. Both have sufficient volume to implement such a strategy. The limited volume of CubeSats and PocketQubes typically prohibits such a redundancy concept. It may however be possible to implement such a concept on the same printed circuit board, provided that it fits, but not on the full satellite. On Delfi-n3Xt, for example, this was implemented on the partially redundant attitude determination and control subsystem for the magnetorquers as shown in 4.10.

The available statistical information from the CubeSat failure database is on a limited set of subsystems and already includes the interfaces in the relevant subsystem as explained in Section 5.6. This is considered to sufficient for reliability modelling at satellite level when reliability analysis is limited to full subsystem redundancy using identical physical subsystem units and electrical interfaces. For selective redundancy or for redundancy using a different unit as secondary system, failure distributions for the real physical subsystems are required.

5.7.2. Unit Dependence of Redundant Subsystems

Definitions for dependent failures can be confusing. In Table 5.4 an overview of definitions is provided, where the vertical bars on the left indicates that the underlying failure type(s) are a subset of the above.

Cascade failures are ignored in this study as only fatal failures are considered and all subsystems are assumed to be critical. For the other dependent failures,

Table 5.4: Definitions of dependent failures as adopted from Borcsok [27].

dependent failure	The likelihood of a set of events, the probability of which cannot be expressed as simple product of unconditional failure probabilities of the individual events.
common cause failure	This is a specific type of dependent failure that arises in redundant components where simultaneous (or near simultaneous) multiple failures result in the same way or in different ways from a single shared cause.
common mode failure	This term is reserved for common-cause failures in which multiple items fail in the same way.
cascade failure	These are all those dependent failures that have no common cause, i.e. they do not affect redundant components.

several methods are available to model the failure dependency between redundant units. Examples are the basic parameter model, the alpha factor model and beta factor model [28]. However, they can be applied only if the dependency is time-independent. An approach is required to address dependent failures for redundant systems over time. Two dependencies of the secondary unit on the primary unit are considered: the start of the lifetime at risk and the time-to-failure dependency.

5.7.3. Lifetime-at-risk Dependency

The start of the lifetime at risk of the secondary unit depends on whether the underlying failure causes apply only when the unit is operational or also when the unit is switched off. It also depends on whether the secondary unit is hot- or cold-redundant. For CubeSats however, power consumption is a major issue and therefore it is assumed that only cold redundancy can be considered at subsystem level. Failure causes can be differentiated between environmental effects and operational effects, relating to an external and internal cause respectively. For a cold redundant subsystem, which is assumed for this study, the cumulative environmental effects (such as thermal cycling, ionization, externally induced vibration, UV, etc.) affect both units from the time the satellite is deployed into orbit, regardless of its operational state. The redundant unit may in principle already have failed before it is commanded to be turned on. Most single environmental effects (such as latch-up or bit upsets) only applies when the unit is active. The same holds for all operational effects. The parameter ϵ is introduced as the probability of a failure dependent on orbital lifetime, independent of its operational state. This yields that $1 - \epsilon$ is the probability of failure dependent on operational lifetime. The lifetime of the cold redundant secondary unit in this case starts after the primary has failed.

5.7.4. Time-to-failure Dependency

The second dependency relates to the time-to-failure for a redundant subsystem. If the redundant subsystem comprises identical units, a subset of common cause and common mode failures related to design flaws (causing immaturity failure) yields a time-to-failure dependency of secondary unit to the primary. If, for example, an electrical power subsystem fails after one orbit because the battery could not handle the peak temperature as it was never designed for that temperature or the thermal analysis was flawed, it becomes very likely that an identical redundant unit also fails in approximately the same time span. The same is true if it survives for years. Some immaturity failure root causes are however independent between the units of a redundant subsystem. Failures in component production or assembly of random nature are considered in the scope of immaturity failures but do not yield dependencies between identically designed units. The beta factor model [28] can be adopted to account for the ratio of lifetime dependent failures β for immaturity failures, with $\beta = 0$ for fully lifetime independent and $\beta = 1$ for fully dependent failures. While the initial $f(t)$ used for the primary subsystem has a decreasing probability density over time, the updated $f(t)$ is expected to have a narrow log-normal distribution with its mode (peak density) around the failure time of the primary subsystem. The exact parameters of this distribution are however unknown. Assuming a narrow distribution which is approximately symmetric around the mode, the failure time of the secondary unit can be approximated by that of the primary.

When the primary unit fails due to wear-out, this is due to accumulative effects for which failures typically have a high variance. While the distribution of a specific type of wear-out can differ from the overall wear-out distribution, the time-to-failure dependency for a secondary unit is limited and unknown. Therefore, the original probability density used for the primary unit can best be applied to the secondary as well and a potential time-to-failure dependency is ignored.

5.7.5. Modelling of Failure Dependencies

The database from the CubeSat survey [29], which is used as input for the CubeSat failure database [2], comprises confirmed or expected root causes of satellite and/or mission failure for 30 of 60 launched CubeSats. Using this input, the cause has been classified in terms of time-to-failure-dependence (β) and its lifetime-at-risk dependence (ϵ) in case a hypothetical identical redundant unit would have been applied. Regarding time-to-failure this yields 12 dependent, 6 independent and 12 unknown failures. Regarding lifetime-at-risk dependence, this yields 13 failures related to orbital lifetime, 10 related to operational lifetime and 7 unknown failures. The problem can be approached as two Bernoulli experiments for β and ϵ, where a 'success' relates to cases which confirm time-to-failure and orbital lifetime dependence respectively and a 'failure' relates to cases which are time-to-failure independent and dependent on operational lifetime, respectively. When applying Bayesian inference on each parameter, the beta distribution can be used as conjugate prior with hyper-parameters a and b [30]. This means that the prior and posterior are both a beta probability density distribution f_{beta}, provided in Equation 5.48 with gamma function $\Gamma(z)$ as defined by Equation 5.11, a for the number of

successes and b for the number of failures.

$$f_{beta} = \frac{\Gamma(a) + \Gamma(b)}{\Gamma(a)\Gamma(b)} \cdot x^{a-1}(1-x)^{b-1} \tag{5.48}$$

Using $a_{prior} = 1$ and $b_{prior} = 1$ yields a uniform prior over the range [0,1], The number of 'successes' and 'failures can be added respectively to obtain the posterior. The classification according to the survey, however, yields numbers of unknown dependencies. Ignoring them would yield a too strong posterior. Therefore the ratio of classified to total failures n_{tot} is applied as weight factor on the classified failures as provided in Equations 5.49 and 5.50. This yields $a_\beta = 8.2$, $b_\beta = 4.6$, $a_\epsilon = 11.0$ and $b_\epsilon = 8.7$ for which the results are shown in Figure 5.28. For the simulation, samples from these posteriors will be drawn.

$$a_{post} = a_{prior} + \frac{n_s + n_f}{n_{tot}} n_s \tag{5.49}$$

$$b_{post} = b_{prior} + \frac{n_s + n_f}{n_{tot}} n_f \tag{5.50}$$

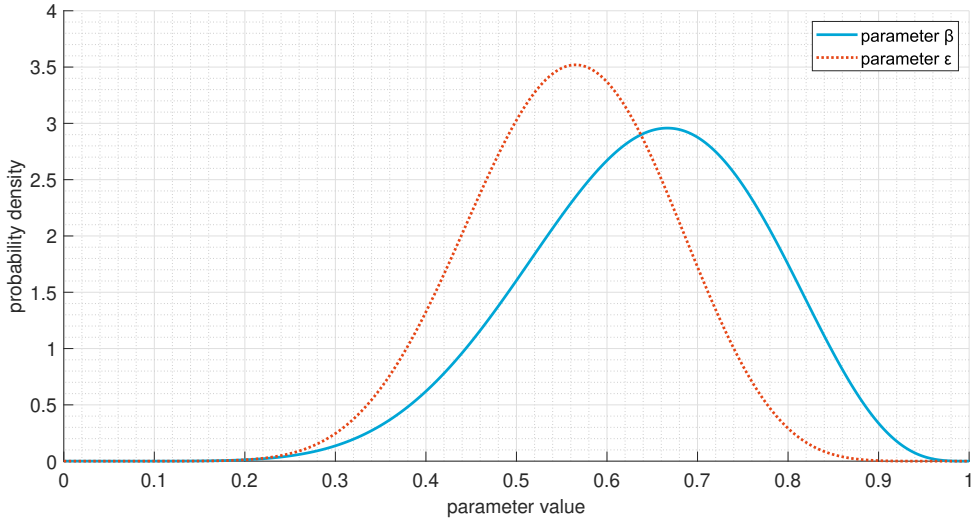


Figure 5.28: Posterior distribution of dependency parameters β and ϵ.

5.7.6. Modelling of Immaturity Failure Mitigation

The reduction of the immaturity failures for a satellite without subsystem redundancy is investigated for the case that project resources, otherwise required for implementing subsystem redundancy, are allocated to measures which reduce immaturity failures instead, e.g. increased testing. In a statistical study of CubeSats, Swartwout states that early failures of CubeSats developed at universities can

mainly be attributed to insufficient or even complete lack of functional system level testing [31]. From the CubeSat survey it follows that the average duration of testing at fully integrated system level of CubeSats is approximately two months [24].

When applying an extensive test campaign to the system for a period of six months, including fixes and improvements where necessary, it is expected that immaturity failures can be reduced significantly. Some critical issues only appear after all subsystems have been integrated, even when subsystems are already tested and found to be fully compliant to its requirements. Therefore it is assumed that most gain in reliability can be achieved by a more extensive test campaign on on a fully integrated satellite. These tests should include long duration testing, professional environmental tests and state-based testing following Failure Mode and Effect & Criticality Analysis (FMECA) for as far as a failure can be detected, isolated and recovered. An example of a 'short' two-month program and an 'extended' half-year program is provided it Tables 5.5 and 5.6 respectively. The short program example is based on experience with the Delfi-n3Xt and Delfi-C^3 satellite. For both programs it is assumed that subsystems have already passed at least functional testing separately by the manufacturer or developer. All tests should be performed on the flight model. However, the functional and duration testing can be done on an engineering model first to identify key design issues while limiting risks and wear of the flight model.

Table 5.5: Example of a limited system level test program for CubeSats.

Test Item	Short Program	Days
functional (nominal)	checking all functions once	15
functional (failure states)	cases commanded by software	15
duration	"day-in-a-life"	2
vibration	only acceptance	2
thermal cycling	required for acceptance	2
thermal compliance	system level hot vacuum	2
bake-out	required for acceptance	1
deployment	once or twice	2
communication	in normal/clean room	2
electrical power	using IV-curve emulator	2
design iteration	"quick & dirty"	10
complete program		55

The reliability can be further improved if a satellite platform is launched, tested in-orbit and subsequently iterated based on the operational lessons learnt. To model these cases of 'improved testing' and 'iterative development', the failures described in Section 5.7.5 are analyzed for this purpose. All of these failures can be classified as immaturity failures. For each failure the likelihood that improved testing and iterative development respectively would mitigate the failure is approximated. This is done in coarse steps of 0.25, with 1 for almost certainly mitigated and 0 for almost certainly not mitigated. The sum of the likelihoods for the 30 analyzed satellites yields an expected mitigation of 16 and 26.5 satellite failures for the improved test-

Table 5.6: Example of an extensive system level test program for CubeSats.

Test Item	Extended Program	Days
functional (nominal)	repetitive, including all possible mode switches	30
functional (failure states)	simulated using dedicated hardware	30
duration	at least three sessions of increasing duration	40
vibration	qualification + acceptance	4
thermal cycling	full thermal vacuum cycling	4
thermal compliance	system level hot vacuum	4
bake-out	according to NASA/ESA standards	2
deployment	at least five times	5
communication	field/anoechic chamber testing	3
electrical power	end-to-end using solar simulator	4
design iteration	full fix including partial re-testing	50
complete program		176

ing and launched iteration cases respectively with respect to the reference case. Beta distributions for the likelihood parameter of subsystem immaturity failure mitigation $P_{imp.}$ for the improved testing case and $P_{iter.}$ for the iterative development case can be constructed in similar fashion to the dependency parameters as explained in Section 5.7.5. Using a uniform prior, the distribution inputs become $a_{imp.} = 17$, $b_{imp.} = 15$, $a_{iter.} = 27.5$ and $b_{iter.} = 4.5$. The results are shown in Figure 5.29.

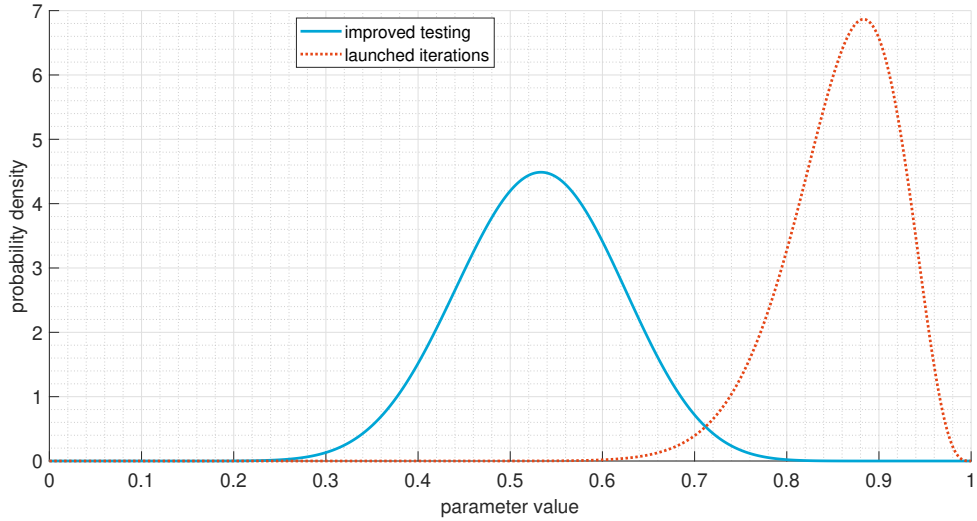


Figure 5.29: Posterior distribution of mitigation parameter $P_{imp.}$ and $P_{iter.}$.

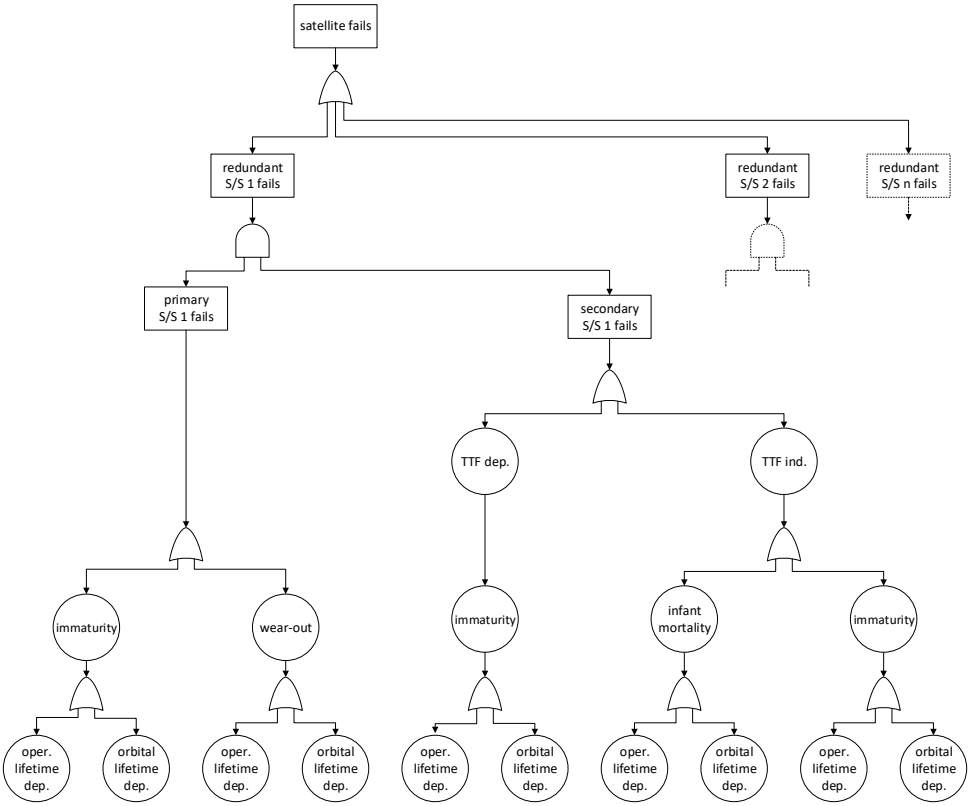


Figure 5.30: Fault tree of a satellite with redundant subsystems.

5.7.7. Satellite Modelling

In Figure 5.30 a fault tree of the failures is provided with differentiation in main failure classes and the aforementioned failure dependencies. The time-to-failure (TTF) dependence (dep.) or independence (ind.) and the operational or orbital life time dependence each lead to different branches. The main branches of the tree are truncated beyond the first subsystem because the branches are identical. The fault tree is a limited representation for a time-dependent model as it does not include the distribution over time and the impact of the dependencies on this distribution. In fault trees, reliability values are typically limited to mutually independent failure probabilities at a given time. The use of dynamic fault trees [32] or Petri-nets [33] could be considered for this purpose as these modelling tools provide options to introduce the impact of these dependencies. However, these tools are focused on multi-level failures which is not in the scope of this study. For the intended model, their diagrams are not as easy to interpret, so a more simple representation is desired. For this purpose, a 'reliability modelling flow' is proposed as a new type of representation for dependent binary-state failures. It is based on an Universal Modelling Language (UML) activity flow.

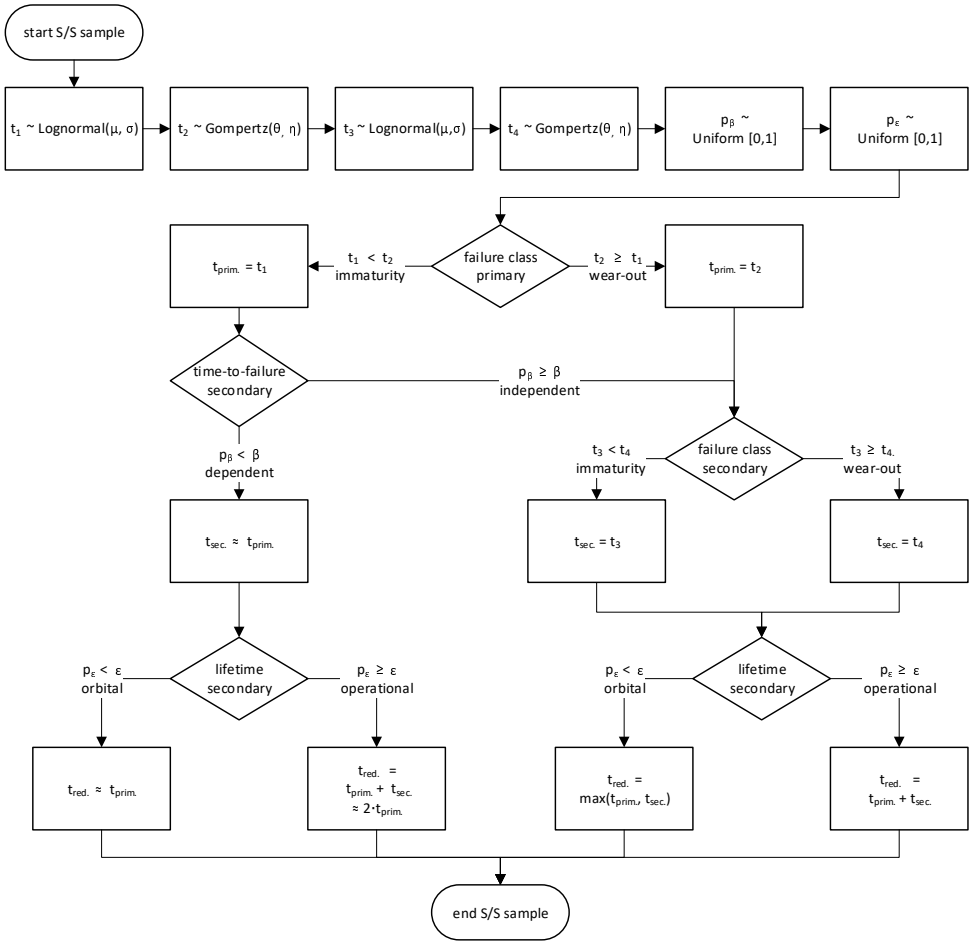


Figure 5.31: Reliability modelling flow for a (cold redundant) subsystem.

The model for a subsystem failure is provided in Figure 5.31. This flow uses sampled input parameters for the subsystem Lognormal-Gompertz product PDF (μ, σ, θ, η) and the dependencies (β, ϵ). It first creates intermediary failure time values t_1 to t_4 for the PDF and reference probabilities p_β and p_ϵ for the dependencies. The failure time values t_1 to t_4 can be generated by using a random generator for a uniform distribution over [0, 1] to create samples for $F(t)$ which are subsequently put into the inverse transform of $F(t)$. An example for the Gompertz distribution is provided by Equation 5.51. To generate a sample for the Lognormal-Gompertz product, the minimum of the samples of the Lognormal distribution and Gompertz distribution for both units is taken as remaining sample, as can be seen in the flow in Figure 5.31. Using these values and following the dependence decisions in the flow, a sample for the primary unit $t_{prim.}$ and the secondary unit $t_{sec.}$ is calculated and the subsystem failure time $t_{red.}$ for a redundant subsystem is generated as output.

The primary unit sample $t_{prim.}$ is the output for a subsystem without redundancy.

$$t = \theta \cdot \ln\left[1 - \frac{\ln(1 - F(t))}{\eta}\right] \tag{5.51}$$

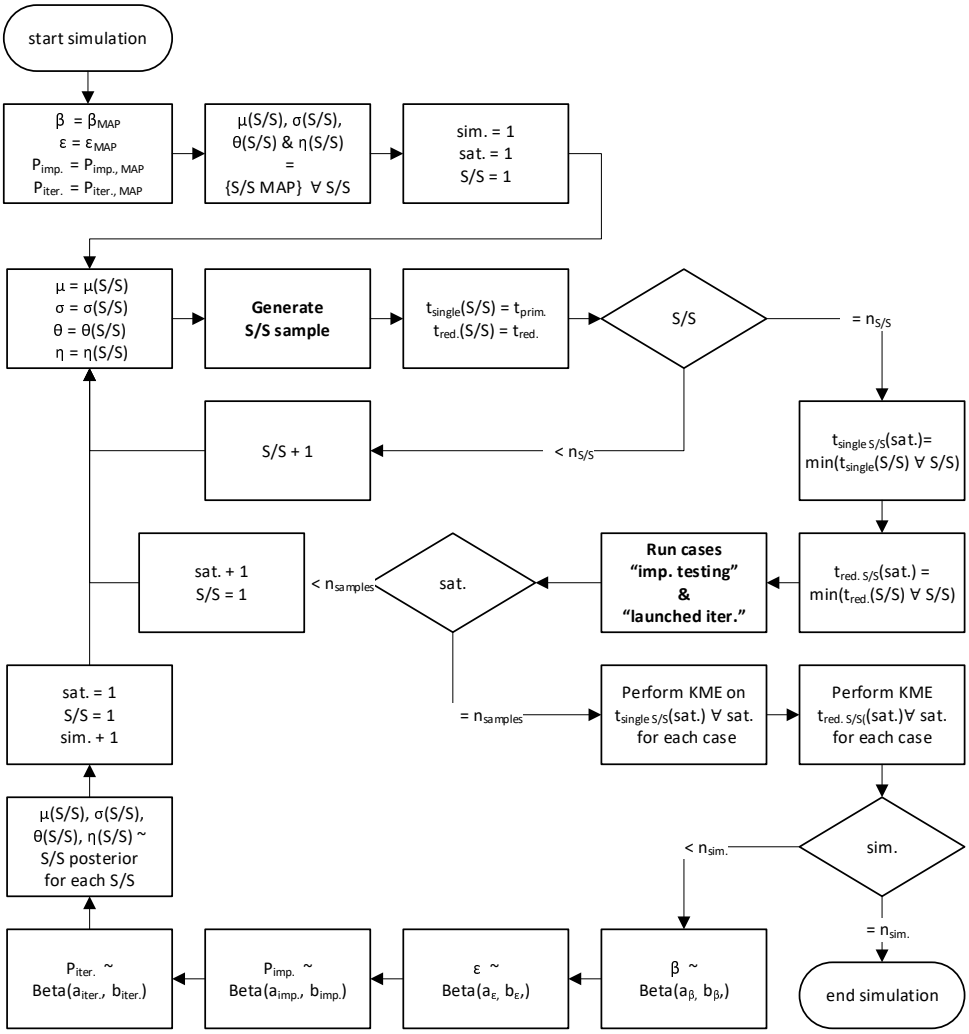


Figure 5.32: Reliability modelling flow for satellite simulation.

Figure 5.32 provides the modelling flow for the reliability of satellites. For all $n_{S/S}$ subsystems, samples are generated. The lifetime of the satellite with redundancy $t_{red.}(sat)$ and without redundancy $t_{single}(sat)$ is modelled as the minimum lifetime of all of its subsystems. A number of $n_{samples}$ satellite samples are generated for each simulation. When enough of such output samples are randomly generated, the

distribution of the output is representative for satellites with and without redundant subsystems. Given a sufficiently high number of samples, a Kaplan Meier Estimator (KME) can be used. Parametric estimates of the output are not needed as they will result to the same figure. A number of n_{sim} simulation runs are performed using the parameter samples from the posterior (see Section 5.6) and samples of the dependency parameters (see Section 5.7.5) to perform a full Monte Carlo simulation which includes the uncertainties on the input parameters. Each simulation results in a different estimated reliability curve.

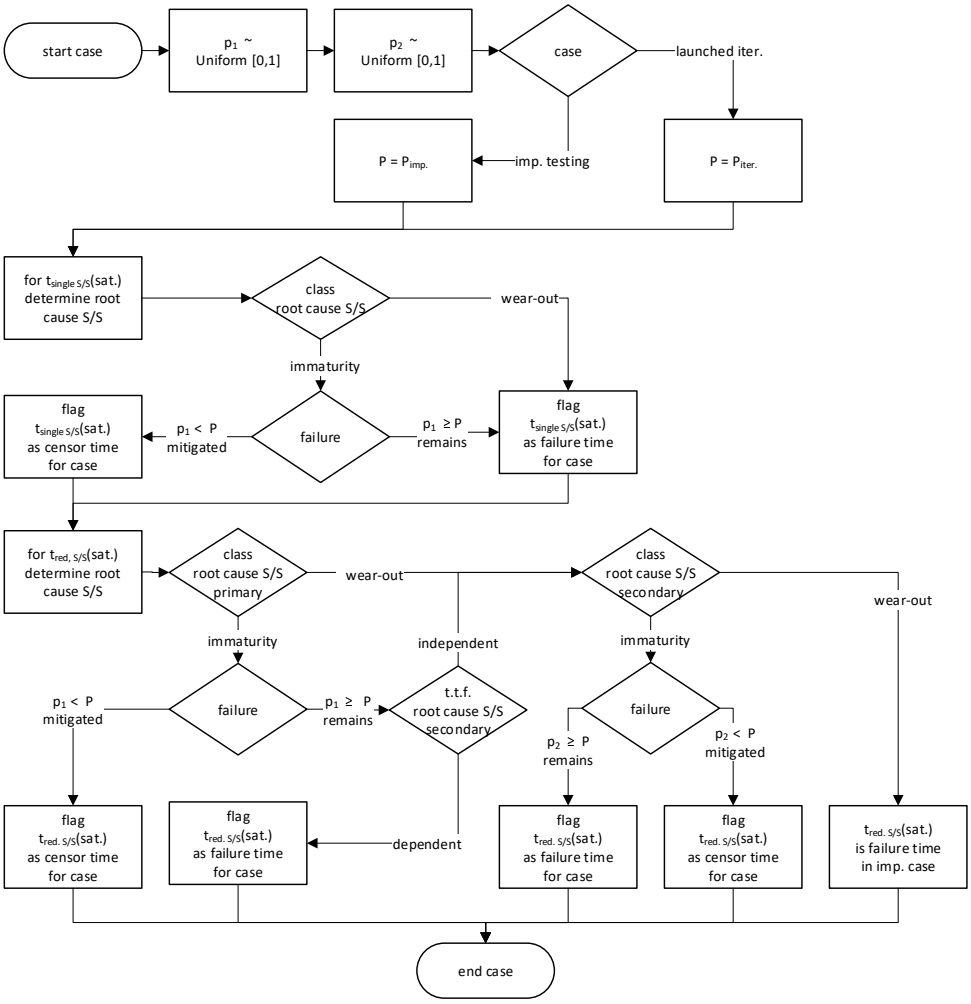


Figure 5.33: Reliability modelling flow for failure mitigation.

In the reference case, all satellite failures are included. For the cases of improved testing and iterative development, root-cause subsystem failures due to immaturity may be mitigated. The activity flow for failure mitigation is provided in

Figure 5.33. Probability values p_1 and p_2 are generated and compared against the samples mitigation parameters $P_{imp.}$ and $P_{iter.}$. The mitigation parameters change for each simulation of $n_{samples}$ and are taken from their posterior beta distributions explained in Section 5.7.6. In case of redundant subsystems, mitigation can only apply to secondaries which fail independently due to immaturity. If the root-cause subsystem failure is mitigated, this failure time is flagged as censor time instead which can be used as input for the KME.

5.7.8. Model Verification

Before running the Monte Carlo simulation on the selected input, the necessary verification steps are performed. The general method applied is to run extreme and/or simplified cases for which the results can be compared with their expected outcomes.

Sample generation from distributions is an essential tool in the simulation. To verify that the sampling method works, 1,000,000 samples are generated using ADCS parameters from Section 5.6. With this number of samples, a histogram can be made as provided in Figure 5.34 using 100 bins which leaves on average 1000 samples per bin. This is deemed more than sufficient to compare it to the original parametric distribution. It shows that the sample generation method produces a proper distribution for the continuous Lognormal-Gompertz distribution. This verification is also performed and proved successfully on the simpler uniform and beta distributions used in the satellite reliability model.

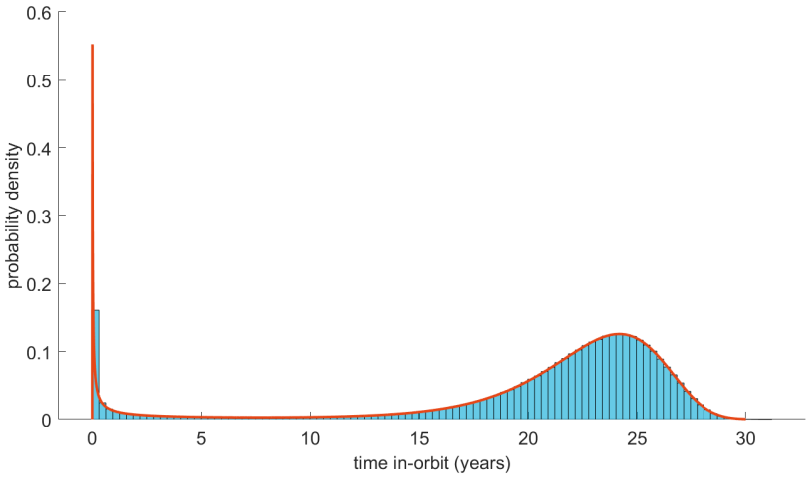


Figure 5.34: Example for Lognormal-Gompertz product sample generation.

Next, the full model, as discussed in Section 5.7.7, is checked. To verify that the dependencies time-to-failure (TTF) and lifetime-at-risk (LaR) work out in the model as expected, parameters are chosen for four theoretical cases where results can be predicted. These cases relate to the four outcomes (result calculation blocks at

the bottom) of Figure 5.31. The distributions for six subsystems are considered to be identical and the same for all simulation runs. The verification cases and the expected outcomes for the reference cases are presented in Table 5.7. For the immaturity failure mitigation cases, it is expected that they have a significant impact for verification cases 1 and 2 and almost no impact for cases 3 and 4.

Table 5.7: Model verification cases.

Dominant Failure Case	**Expected Outcome**
1. immaturity, TTF dependent & orbital LaR	The reliability of a satellite with and without redundant subsystems is almost identical.
2. immaturity, TTF dependent & operational LaR	Most failure times of a satellite with redundant subsystems are doubled w.r.t. a satellite without redundancy. The probability density is stretched accordingly.
3. wear-out, TTF independent & orbital LaR	For a satellite with redundant subsystems, some early failures are mitigated and the probability distribution is squeezed towards the higher failure times w.r.t. a satellite without redundancy.
4 wear-out, TTF independent & operational LaR	Most failure times of a satellite with redundancy are doubled w.r.t. a satellite without and some early failures are mitigated. The probability density is both moved and stretched accordingly.

Table 5.8 provides the input parameters for the verification procedure. The chosen parameters represent the four cases, but they are not chosen as infinitesimally narrow distributions. This is done to prevent that the modelling flow would be followed in one exact path for all samples, which could potentially conceal errors and sensitivities in the model. For the reduction of immaturity failure, the same applies. For all verification cases, the chosen parameters approximate 50% of the immaturity failures mitigated for ”improved testing” and 75% for ”iterative development”.

Table 5.8: Input parameters for verification cases.

	Subsystems				**Dependencies**				**Mitigation Cases**			
Case	**μ**	**σ**	**θ**	**η**	β_a	β_b	ε_a	ε_b	$P_{imp.,\,a}$	$_{,\,b}$	$P_{iter.,\,a}$	$_{,\,b}$
1	-1	1	5	10^{-5}	100	5	100	5	50	50	100	1
2	-1	1	5	10^{-5}	100	5	5	100	50	50	100	1
3	100	5	2	10^{-3}	5	100	100	5	50	50	100	1
4	100	5	2	10^{-3}	5	100	5	100	50	50	100	1

Figure 5.35 and 5.36 provide the sampled distribution of a primary subsystem for cases 1 and 2 and for cases 3 and 4 respectively. Figure 5.37 provides the two sampled dependency distributions.

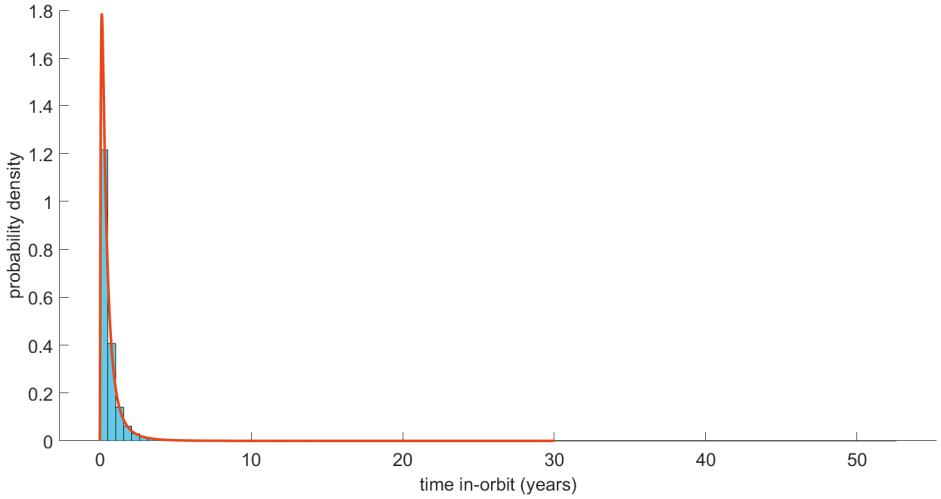


Figure 5.35: Generated samples of primary subsystem for case 1 & 2.

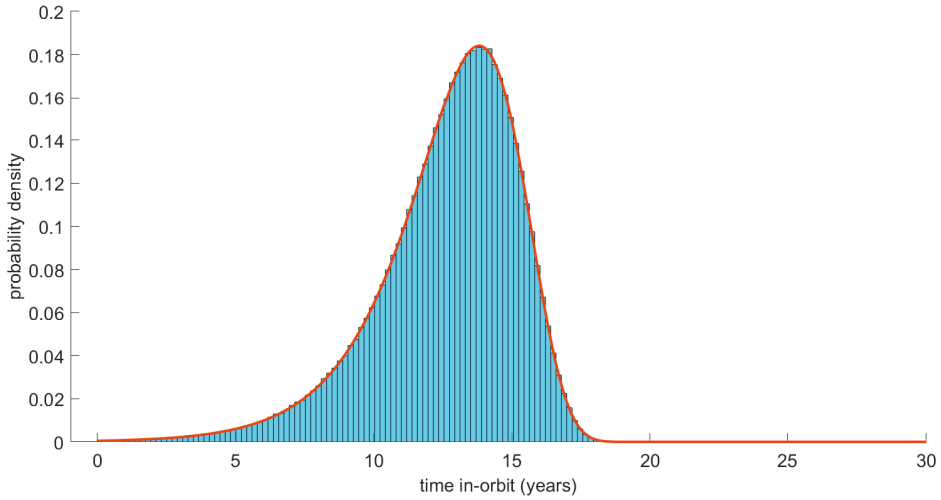


Figure 5.36: Generated samples of primary subsystem for case 3 & 4.

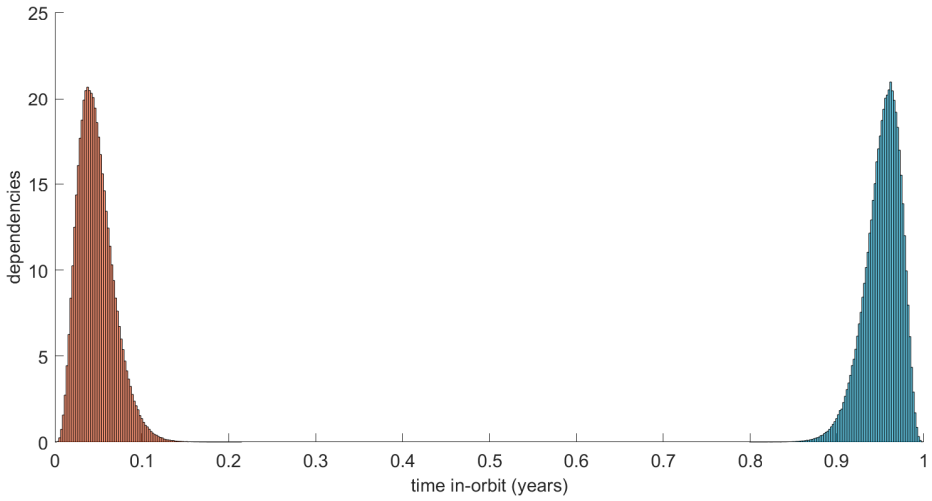


Figure 5.37: Generated samples of dependency parameters.

The simulation is set to 100 runs, each generating 10,000 satellite samples, which is considered to be appropriate for the accuracy of each run and the random distribution of parameter samples. Figures 5.38, 5.39, 5.40 and 5.41 provide the Kaplan-Meier Estimates for the different simulation runs and a histogram of the satellite reference case samples (all runs combined). The results for all verification cases are as expected in relation to the dependency parameters and the immaturity failure mitigation. However, the distribution of verification case 4 shows two peaks in the results which requires some investigation. It turns out that, although the chosen parameters have a dominant wear-out for a single subsystem, the chance that either a primary or secondary of six subsystems fails due to immaturity is still significant. Because of high operational dependence, the time of the other unit of the first failing subsystem is added, which is very likely due to wear-out. This explains the first peak, while the second peak relates to both units failing due to wear-out. Overall, it can be concluded that the model is working correctly.

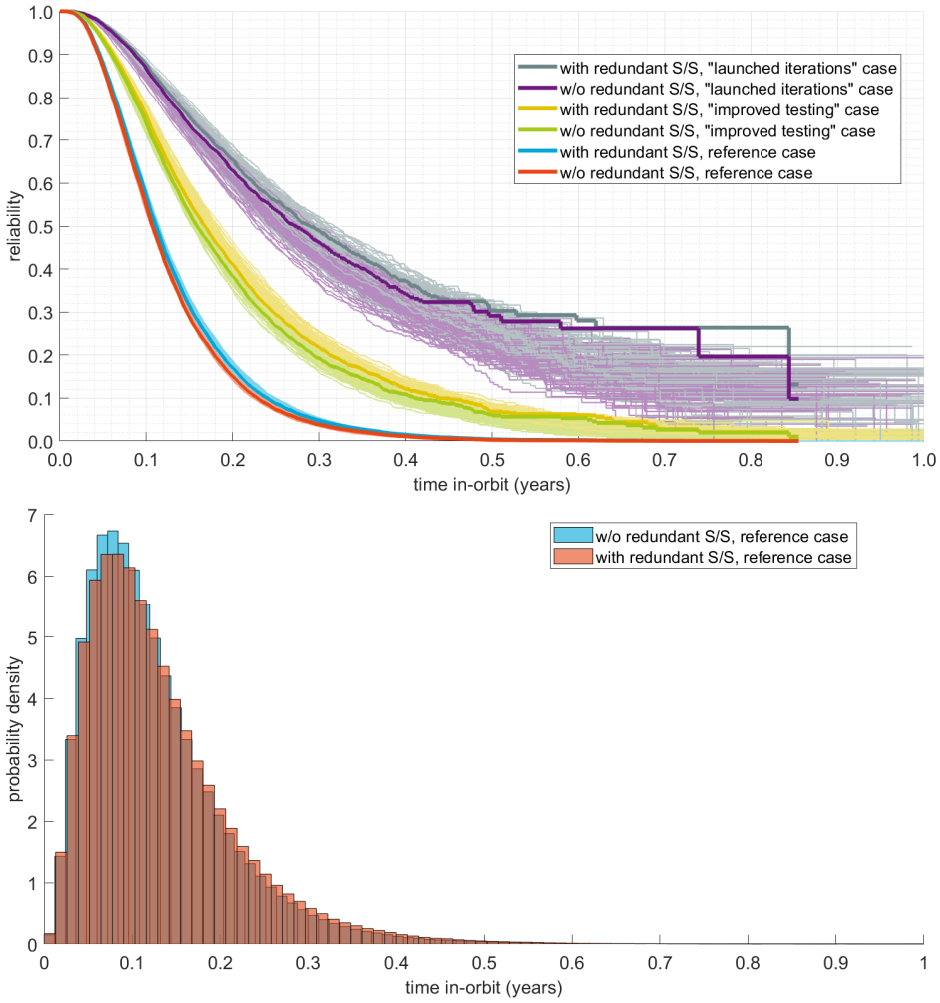


Figure 5.38: KME and histogram of output samples of verification case 1.

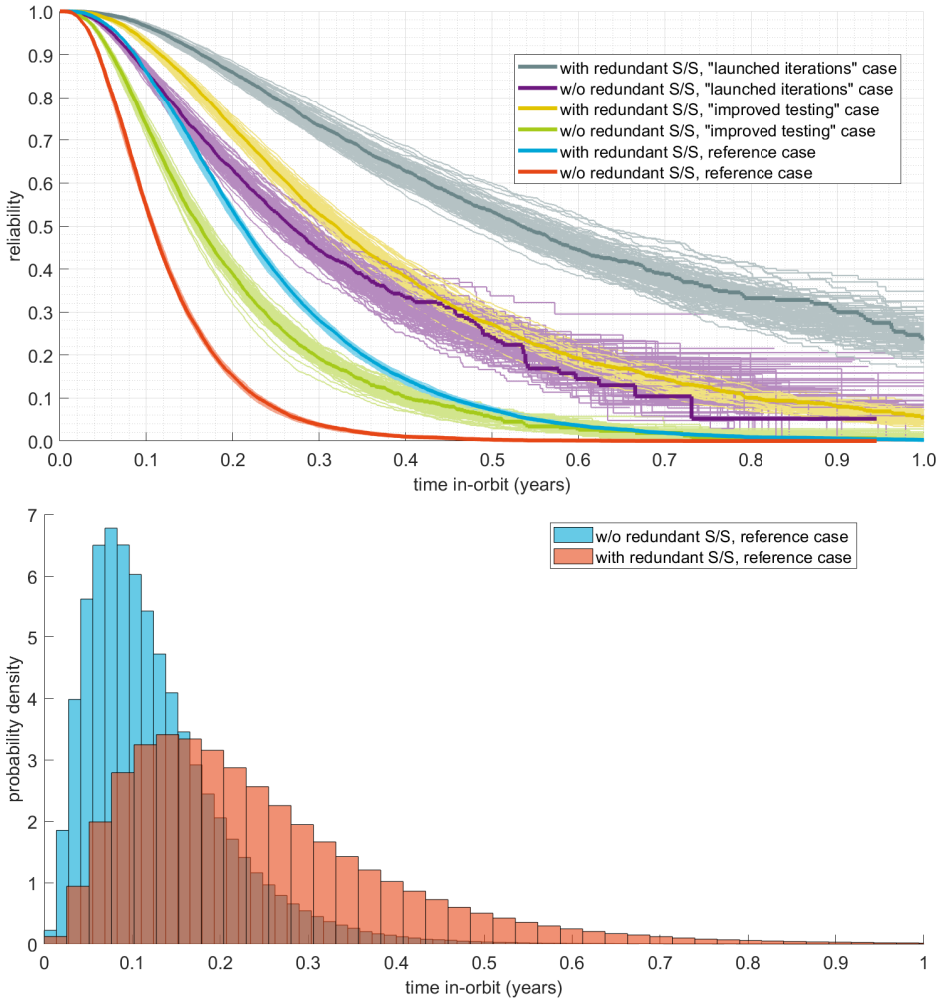


Figure 5.39: KME and histogram of output samples of verification case 2.

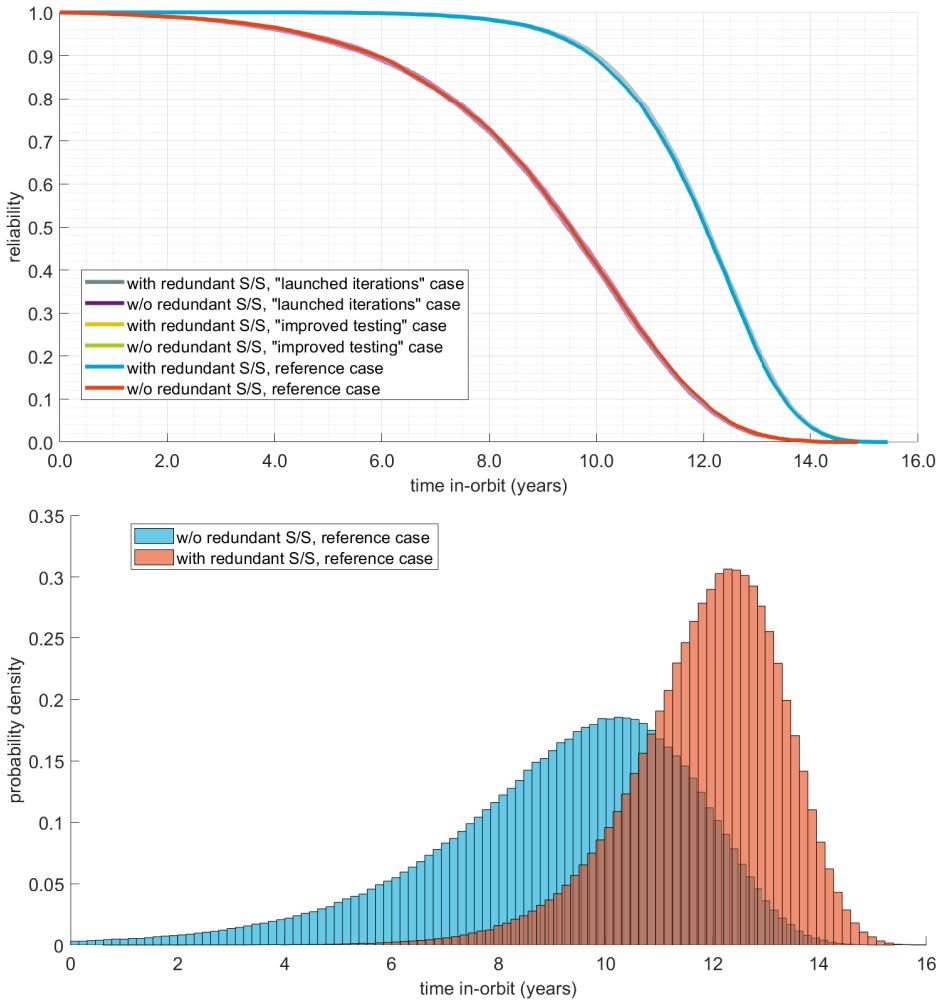


Figure 5.40: KME and histogram of output samples of verification case 3.

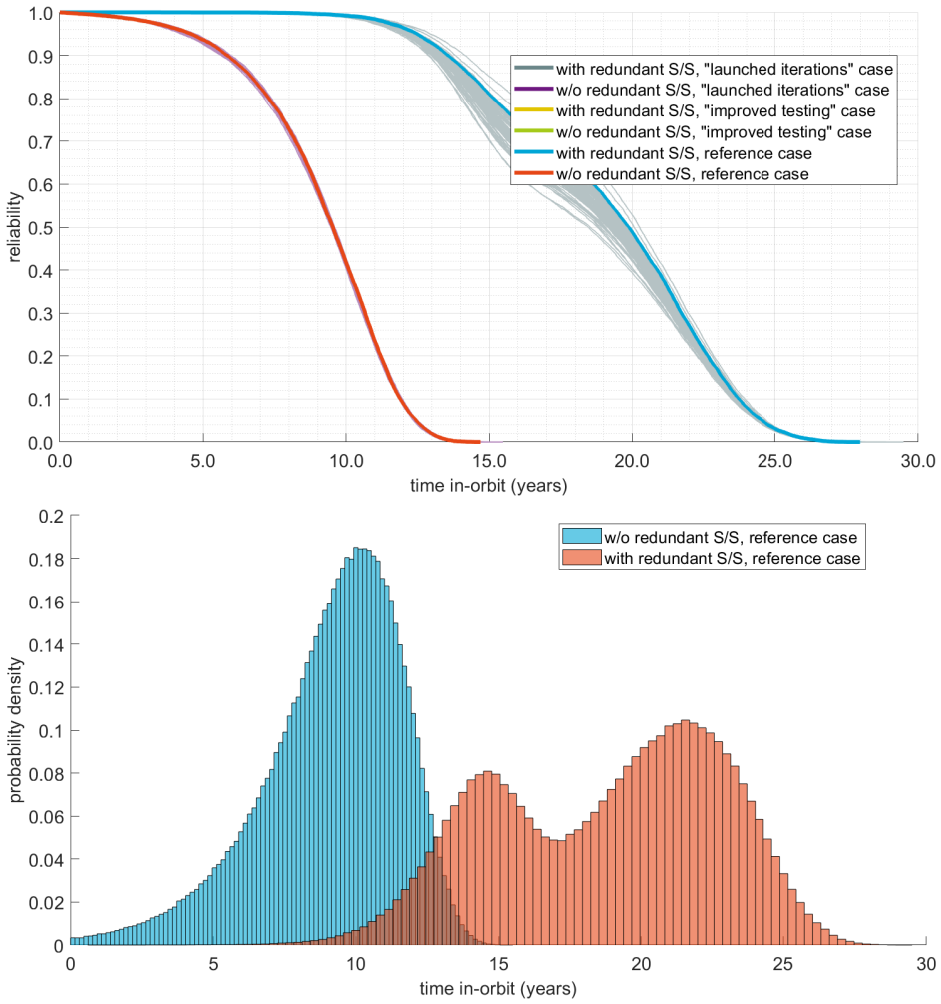


Figure 5.41: KME and histogram of output samples of verification case 4.

5.8. Results of the Satellite Reliability Simulation

The reliability of a satellite with redundant subsystems is simulated in 100 runs, each producing 10,000 samples for satellite failure using the model explained in Section 5.7.7. The results for the reliability over time is provided in Figure 5.42 in the form of a Kaplan Meier Estimate. The thick lines represent the reliability over time using the subsystem MAP estimates as input, while the smaller lines are the results from simulation runs using other samples of the subsystem posterior distributions as explained in section 5.6. In Table 5.9 the reliability after a specified time in orbit is provided for the MAP inputs and the mean of all simulation runs.

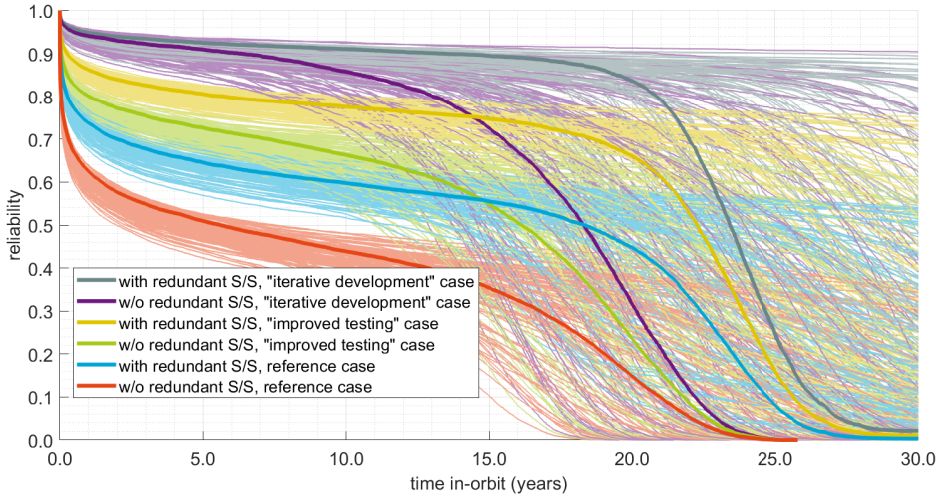


Figure 5.42: Kaplan Meier Estimate simulation results.

Table 5.9: Satellite simulation reliability at specified time in-orbit

S/S red.	case	1 year MAP	1 year mean	3 years MAP	3 years mean	5 years MAP	5 years mean	10 years MAP	10 years mean
w/o	ref.	0.62	0.60	0.54	0.52	0.51	0.49	0.44	0.42
with	ref.	0.74	0.73	0.67	0.67	0.64	0.64	0.60	0.59
w/o	imp.	0.80	0.79	0.75	0.74	0.72	0.71	0.66	0.65
with	imp.	0.86	0.85	0.82	0.81	0.80	0.79	0.77	0.76
w/o	iter.	0.95	0.93	0.93	0.91	0.91	0.89	0.86	0.85
with	iter.	0.95	0.93	0.93	0.91	0.92	0.90	0.91	0.89

Figure 5.42 already clearly shows a ranked order based on the subsystem MAP inputs for the first 15 years in orbit. A similar ranking can be seen in Table 5.9. Based on these outputs, a satellite without redundant subsystems and improved testing seems to perform better than and a satellite with redundant systems in the reference case. The lines for the different simulations, however, partially overlap

and could therefore in specific simulation runs yield a different outcome. For this purpose, the reliability for each simulation run after 1, 3, 5 and 10 years are compared. Most CubeSat design lifetimes will be in the range of one and five years (see Section 5.2.2). The reliability at 10 years is therefore of limited interest, but added for some unique future missions. In Figure 5.43 a scatter plot is provided, with on the horizontal axis the reliability for a satellite with redundant subsystems and on the vertical axis the reliability for a satellite without subsystems but improved testing instead. The diagonal line indicates the theoretical boundary where the reliability of the two options would be equal. This figure confirms that allocating resources to improved testing in general has a better impact on reliability than subsystem redundancy. Only for very long missions of 10 years, there is a small chance that subsystem redundancy is superior to improved testing.

Figure 5.44 compares the CubeSat with and without redundant subsystems for the reference case of both. It clearly shows the positive impact of redundancy. This means that when one chooses to improve testing instead of implementing redundancy, it must be assured that the time and resources are truly secured.

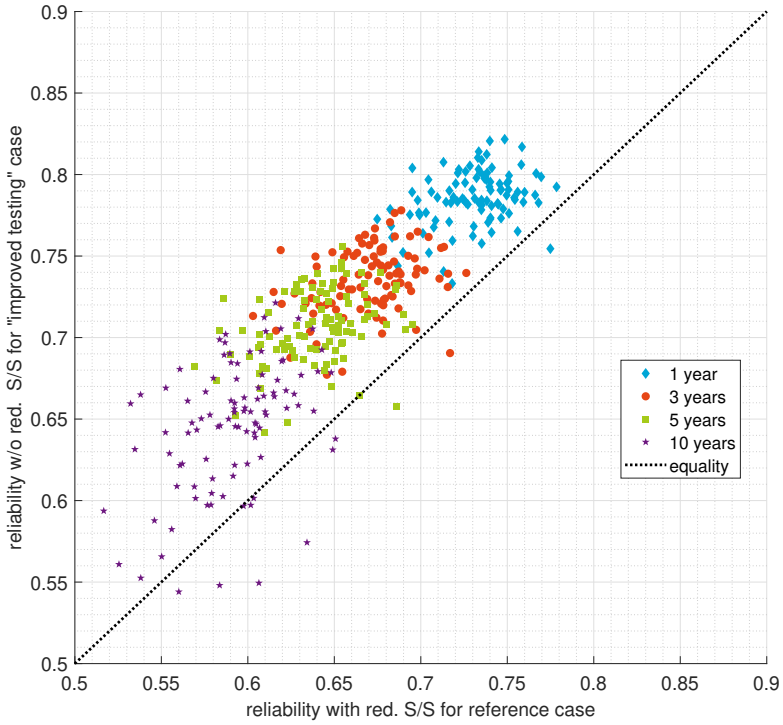


Figure 5.43: Satellite reliability scatter for redundant subsystems in reference case vs non-redundant subsystems with improved testing.

In case there is sufficient time and manpower available to implement both subsystem redundancy and improved testing, this does yield a significant improvement as shown in Figure 5.45.

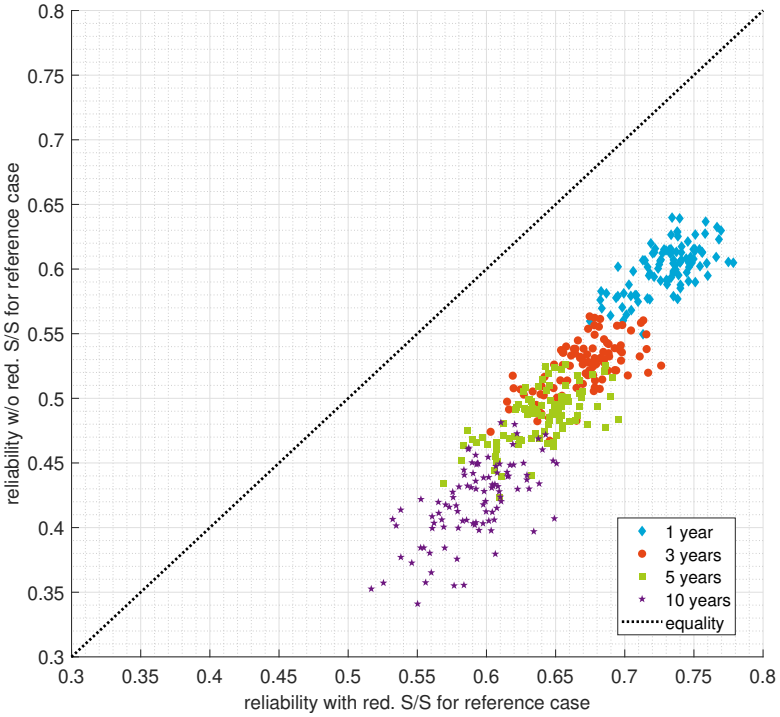


Figure 5.44: Satellite reliability scatter for the reference case.

Finally, it is interesting to compare satellites with and without redundant subsystems in the case of iterative development. This is especially interesting for CubeSat networks or a standardized CubeSat platform for multiple missions. In this case there is not so much of a trade to make based on project resources, as iterations may already been foreseen by programmatic choice. The results are provided in Figure 5.46. For missions up to 3 years, the results are scattered around the equality line. Only for missions of 5 years or more, redundancy does pay off.

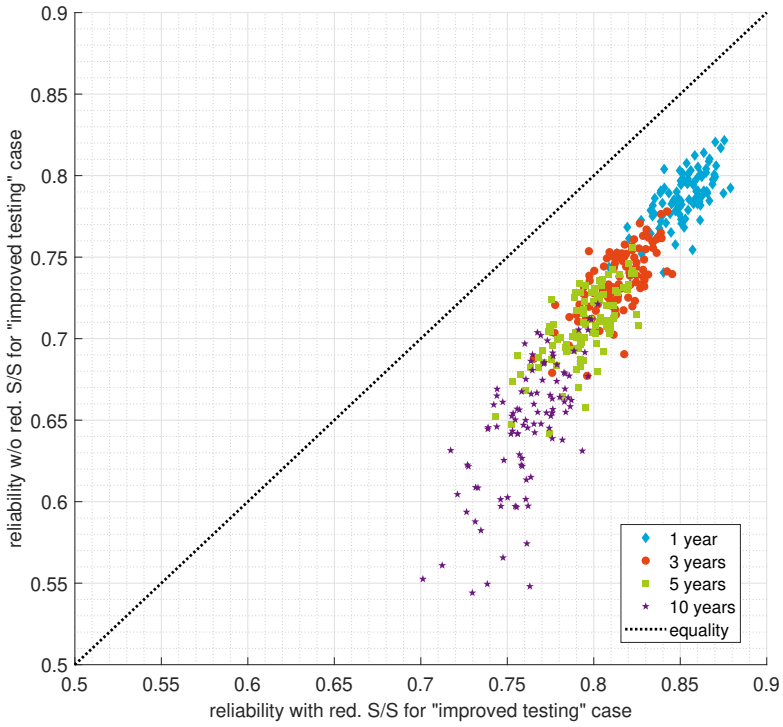


Figure 5.45: Satellite reliability scatter for the ”improved testing” case.

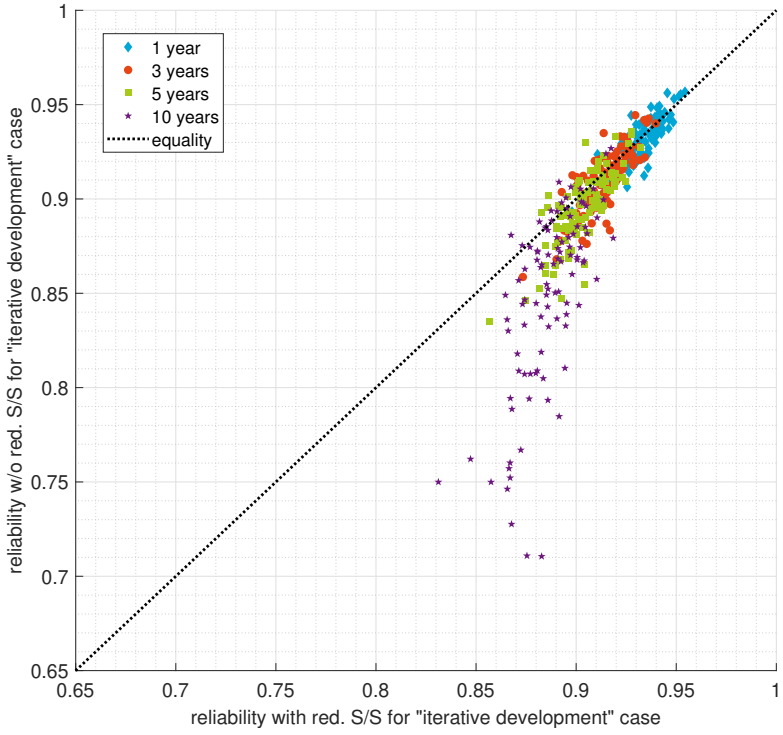


Figure 5.46: Satellite reliability scatter for the ”iterative development” case.

5.9. Conclusions on Reliability

The answer to the question "What leads to higher CubeSat reliability over its mission life time: full subsystem redundancy or improved testing?" is that improved testing yields the best results for most missions, based on the simulation results presented in Figure 5.43. It has the additional benefit to increase potential payload volume and has a lower platform cost compare to a satellite with redundant subsystems. Furthermore, iterative development is the best strategy for series and networks of CubeSats and/or bus platforms. A satellite with redundant subsystems remains more reliable than a satellite without redundancy above 10-15 years as shown in Figure 5.42. This is however beyond the typical useful lifetime of CubeSats and therefore considered to be of limited importance. It can also be seen, by the spread of the simulated curves, the uncertainty of the model increases over time which is mainly due to limited observations causing relatively large uncertainties of the wear-out parameters.

A Lognormal-Gompertz product provides the best parametric reliability model for a CubeSat and small satellites in general. The Kaplan-Meier estimator is a necessary step to show the pure satellite observation data and provide a reference for parametric models. Maximum likelihood estimators are good for model comparison, but lack the ability to introduce prior knowledge. Furthermore, they do not provide a full posterior which limits the ability to properly model uncertainties in subsequent modelling. Bayesian inference is the best approach to overcome these limitations. This paper provides an example of the use of these tools, which can be of interest to satellite reliability modelling or even reliability modelling in general. The satellite model as introduced in Section 5.7.7 is an innovative method which can be applied to other satellite size categories as well as other complex systems.

The research question implies a choice between allocating additional resources to either the implement redundant subsystems or to improve testing on a satellite without redundancy. Such a choice may not be applicable if resources allow for both improvements. In this case, the reliability of a satellite with redundancy has a significantly higher reliability over time compared to a satellite without. However, applying redundancy does not only consume additional organizational resources. It also consumes a considerable amount of volume of the satellite which leaves less room for the payload. Moreover, this study assumes a flawless failure detection, isolation and recovery mechanism which arbitrates between the redundant units of a subsystem which may be too optimistic in reality. For a single satellite in a project with limited resources, it is therefore considered to be a better strategy to aim for reduction of immaturity failures through extensive testing. For satellite networks, 'swarm robustness' could be achieved [1] and individual satellite losses may be acceptable. Only for single satellite missions longer than 10 years or operating in harsher environments beyond LEO, redundant subsystems may be required to improve reliability to acceptable values.

Immaturity failures can be reduced by iterating on the satellite bus platform with subsequent launches and high reliability can be achieved, as shown in Figure 5.46. This would extend improved on-ground testing to in-orbit testing. A condition for this approach is that improvements in performance of each subsequent

design remain limited and improvements are primarily focused on the reliability of the design. A modular philosophy of CubeSats where subsystems from different manufacturers are procured and integrated is not compatible with this strategy as the lack of direct involvement may prohibit the required improvement or the selection of a different model can introduce new immaturity failure risks at subsystem or satellite level. The entire iterative platform development must therefore be under control of one party or consortium. While this may have its limitations, it opens the possibility to introduce advanced architectural concepts which deviate from the modular approach, as introduced in Chapter 4. An example of such architecture is the integration of satellite core functionality into a single physical unit to reduce its effective volume and reduce the component count (potentially increasing reliability further). Another example is the used of advanced outer panels which reduce wiring harness and integration complexity. A disadvantage of an integrated approach by major entities with a high focus on reliability is that it may slow down innovation and the access of new (small) players. To allow disruptive innovation and new players, a relatively reliability with higher immaturity failure for novel CubeSat and/or new players has to be accepted.

References

[1] S. Engelen, E. Gill, and C. Verhoeven, *On the Reliability, Availability, and Throughput of Satellite Swarms,* IEEE Transactions on Aerospace and Electronic Systems **50**, 1027 (2014).

[2] M. Langer and J. Boumeester, *Reliability of CubeSats – Statistical Data, Developers' Beliefs and the Way Forward,* in *Proceedings of 30th AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2016).

[3] *ECSS-Q-ST-30C Rev.1: Dependability*, Tech. Rep. 3rd, rev.1 (European Cooperation for Space Standardization, 2017).

[4] G. A. Klutke, P. C. Kiessler, and M. A. Wortman, *A Critical Look at the Bathtub Curve,* IEEE Transactions on Reliability **52**, 125 (2003).

[5] K. L. Wong, *The Physical Basis for the Rollercoaster Hazard Rate Curve for Electronics,* Quality and Reliability Engineering International **7**, 489 (1991).

[6] J. Guo, L. Monas, and E. Gill, *Statistical Analysis and Modelling of Small Satellite Reliability,* Acta Astronautica **98**, 97 (2014).

[7] E. L. Kaplan and P. Meier, *Nonparametric Estimation from Incomplete Observations,* Journal of the American Statistical Association **53**, 457 (1958).

[8] J.-F. Castet and J. H. Saleh, *Satellite Reliability: Statistical Data Analysis and Modeling,* Journal of Spacecraft and Rockets **46**, 1065 (2009).

[9] J. F. Castet and J. H. Saleh, *Satellite and Satellite Subsystems Reliability: Statistical Data Analysis and Modeling,* Reliability Engineering and System Safety **94**, 1718 (2009).

[10] M. Greenwood, *A Report on the Natural Duration of Cancer.* Reports on Public Health and Medical Subjects (Ministry of Health) **33** (1926).

[11] J. F. Castet and J. H. Saleh, *Single versus mixture Weibull distributions for nonparametric satellite reliability,* Reliability Engineering and System Safety **95**, 295 (2010).

[12] EPSMA, *Guidelines to understanding reliability prediction,* Report (2004).

[13] X. Liu, *Survival Analysis: Models and Applications* (2012).

[14] E. T. Lee and O. T. Go, *Survival Analysis in Public Health Research,* Annual Review of Public Health **18**, 105 (1997).

[15] W. Weibull, *A Statistical Distribution Function of Wide Applicability,* Journal of Applied Mechanics **18**, 293 (1951).

[16] B. Gompertz, *On the Nature of the Function Expressive of the Law of Human Mortality, and on a New Mode of Determining the Value of Life Contingencies.* Proceedings of the Royal Society of London **2** (1833), 10.1098/rspl.1815.0271.

[17] E. E. Elmahdy and A. W. Aboutahoun, *A New Approach for Parameter Estimation of Finite Weibull Mixture Distributions for Reliability Modeling,* Applied Mathematical Modelling **37**, 1800 (2013).

[18] M. Xie, Y. Tang, and T. N. Goh, *A Modified Weibull Extension with Bathtub-Shaped Failure Rate Function,* Reliability Engineering and System Safety **76**, 279 (2002).

[19] W. Peng, H. Zhang, H. Z. Huang, Z. Gong, and Y. Liu, *Satellite Reliability Modeling with Modified Weibull Extension Distribution,* in *Proceedings of 11th International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering* (Chengdu, 2012) pp. 195–199.

[20] T. Bayes and R. Price, *An Essay Towards Solving a Problem in the Doctrine of Chances,* Philosophical Transactions (1683-1775) **53**, 370 (1763).

[21] R. G. D. Steel and J. H. Torrie, *Principles and Procedures of Statistics with Special Reference to the Biological Sciences.* (McGraw-Hill, 1960).

[22] D. M. Miles, I. R. Mann, M. Ciurzynski, D. Barona, B. B. Narod, J. R. Bennest, I. P. Pakhotin, A. Kale, B. Bruner, C. D. Nokes, C. Cupido, T. Haluza-DeLay, D. G. Elliott, and D. K. Milling, *A Miniature, Low-power Scientific Fluxgate Magnetometer: A Stepping-stone to CubeSatellite Constellation Missions,* Journal of Geophysical Research: Space Physics **121**, 11839 (2016).

[23] K. Aho, D. Derryberry, and T. Peterson, *Model Selection for Ecologists: The Worldviews of AIC and BIC,* Ecology **95**, 631 (2014).

[24] M. Langer, *Reliability Assessment and Reliability Prediction of CubeSats through System Level Testing and Reliability Growth Modelling*, Ph.D. thesis, Technical University of Munich (2018).

[25] J. Eickhoff, *The FLP Microsatellite* (Springer, 2016).

[26] T. Bayer, *Planning for the Un-plannable: Redundancy, Fault Protection, Contingency Planning and Anomaly Response for the Mars Reconnaissance Orbiter Mission,* in *Proceedings of the Space Conference & Exposition* (AIAA, Long Beach, 2007).

[27] J. Börcsök, S. Schaefer, and E. Ugljesa, *Estimation and Evaluation of Common Cause Failures,* in *Proceedings of the 2nd International Conference on Systems* (2007).

[28] H. W. Jones, *Common Cause Failures and Ultra Reliability,* in *Proceedings of the 42nd International Conference on Environmental Systems* (2012).

[29] J. Bouwmeester, M. Langer, and E. Gill, *Survey on the Implementation and Reliability of CubeSat Electrical Bus Interfaces,* CEAS Space Journal **9** (2017), 10.1007/s12567-016-0138-0.

[30] D. Fink, *A Compendium of Conjugate Priors*, Tech. Rep. (Montana State University, Bozeman, 1997).

[31] M. Swartwout, *The First One Hundred CubeSats : A Statistical Look,* Journal of Small Satellites **2**, 213 (2013).

[32] V. Volovoi, *Modeling of System Reliability Petri Nets with Aging Tokens,* Reliability Engineering and System Safety **84**, 149 (2004).

[33] M. Čepin and B. Mavko, *A Dynamic Fault Tree,* Reliability Engineering and System Safety **75**, 83 (2002).

6

Conclusions and Recommendations

We have the duty of formulating, of summarizing,
and of communicating our conclusions,
in intelligible form,
in recognition of the right of other free minds
to utilize them in making their own decisions

Ronald Fisher

In this chapter a summary of the key results is provided. This is followed by general conclusions, highlights as well as recommendations for future research. Finally, an outlook is provided for the implementation of this research and the developments of CubeSats and PocketQubes in general.

6.1. Summary of Results

This section presents a summary of the results. The results are structured as answers to the research questions which are provided in the Introduction (1.3). The overarching research question "*Which satellite bus architecture provides a reliable solution to the needs and constraints of a CubeSat and a PocketQube mission?*" has been broken down in two high level research questions which will be treated in Sections 6.1.1 and 6.1.2 respectively.

6.1.1. The Impact of Bus Architecture

The main research question "*1. What is the impact of the bus architecture on the reliability and performance of a CubeSat and PocketQube?*" has been broken down in three sub-questions.

1.a) What is the overall reliability of launched CubeSats? Which issues with the bus architecture can be identified and what is their relative impact on the overall reliability?

A survey on CubeSats has been performed by distributing a questionnaire to the CubeSat community in November 2014. After processing the responses, results for 60 launched CubeSats and 44 CubeSats in development have been analyzed. The launched CubeSats in this survey are all between 1U and 3U in size. They account for 24% of the total launched CubeSats at the time the questionnaire was sent out. According to this survey, only about one-third of the CubeSats missions have achieved full mission success in 2014 (Section 1.1). On the other hand, 90% of the satellites have been operational for at least a day after they have been injected into orbit, 80% even for at least their design lifetime. The vast majority of missions has multiple high-level mission objectives (e.g. education, technology demonstration, science, etcetera) and for most of them, at least one of these objectives is met. The electrical power subsystem is responsible for at least 11% of in-orbit failures causing a reduction of mission success (Section 2.2.2). Two out of five CubeSats which did not implement any current protection on the power distribution lines have failed after a few days in orbit (Section 2.4). While data busses are not a major source of in-orbit failures, most CubeSats have implemented I^2C and 60% of them have experienced in-orbit issues such as bus lock-ups (Section 2.3).

1.b). What is the state-of-the art performance of a CubeSat and PocketQube bus and which demand can be foreseen in the near future? What is the impact of the bus architecture on this performance?

For power distribution, some CubeSat architectures include multiple voltage level conversion steps in series which yield relatively high power losses. The dominant data bus used in CubeSats, I^2C, yields limitations on the data rate up to about

250 kbit/s effectively (Section 3.4.2). For housekeeping, this may be sufficient. However, for demanding payloads, high-speed radios and data storage, additional point-to-point data links are advised (Section 3.2.1). Most CubeSats and PocketQubes use a modular stack of printed circuit boards to host the subsystems (Section 4.2). While this, in principle, allows for mixing subsystems from different commercial suppliers, it has been observed that commercial suppliers of CubeSat systems use different power distribution architectures and wiring allocation, even if they use the same PC/104 stack-through interface connector (Section 2.5). For example, one manufacturer distributes a 3.3 V, a 5 V and battery voltage bus, while another manufacturer complements this with 6 different configurable distribution lines using pins on the connector which are not specifically allocated for this. This leads to compatibility issues. Furthermore, this modularization consumes a high amount of volume compared to identified advanced architectures: in the case study of Delfi-n3Xt, a fully modular architecture consumes about half of the spacecraft volume which is twice that of the proposed advanced architecture (Section 4.5). This ratio becomes worse when redundancy for all critical subsystems is implemented.

Given the identified issues with common CubeSat architectures, it is foreseen that future CubeSats require a higher relative volume for payloads and a lean and compliant electrical interface comprising a robust data bus and power efficient architecture.

1.c) Which reliability metrics and which estimation methods can best be used to model a bus architecture given the provided statistical information? What are the results?

The research on innovative architectures is mainly addressing the satellite and subsystem level. A model for reliability over time was desired at both levels to be able to identify the most important aspects of architecture on reliability and to perform subsequent modelling for improvements on these aspects. A parametric reliability model is considered to be the most useful for this purpose.

At the time of study, statistical information on CubeSat failures is available at satellite and subsystem level (Section 2.2 & 5.2) but the low number of launched PocketQubes is insufficient for statistical analysis. The first part of the reliability analysis focused on finding an appropriate probability density function to model small satellite reliability. Investigation in failure classification has resulted in a division between immaturity and wear-out failures. Wear-out failure is a commonly applied term which addresses accumulative effects due to operation and the environment and has an increasing failure rate over time. Immaturity failure is a new term and comprises the infant mortality as well as any other random, deterministic or operations-induced single event failures. It has been found that both infant mortality and all single event failures contribute to a decreasing failure rate over time. Infant mortality suggests early failure (e.g. first fraction of the mission life time) and is thus considered to be too ill-defined to cover for a decreasing failure rate in which the probability density function stretches over a longer period in time.

Weighted mixtures and products of Gamma, Gompertz, Lognormal, Loglogistic and Weibull distributions have been used and their performances have been com-

pared on several criteria. In existing literature, a mixture of two Weibull distributions has been presented as best practice. It was however found that a Lognormal-Gompertz product distribution (Section 5.5) employs the best characteristics overall. Compared to the previously used Weibull mixture, it scores significantly better on the Akaike Information Criterion (-8 for CubeSats) and has a similar goodness-of-fit. More importantly, this model does not exhibit the issue of unrealistically high survival rates on the very long term (e.g. beyond 50 years). Using Bayesian inference and estimates of a small satellite database, parameters for CubeSats and their subsystems have subsequently been estimated and provided (Section 5.5.4 and 5.6).

6.1.2. An Innovative and Reliable Bus Architecture

The main research question *2. Which innovative and reliable bus architecture meets the typical constraints and performance demands of CubeSats and PocketQubes foreseen in the near future?* has been broken down in two sub-questions.

2.a) Which options can be identified for an innovative CubeSat and PocketQube bus architecture which may tackle the reliability and/or performance issues of existing missions?

The research on architectures first focused on the interfaces between subsystems. A lean electrical interface has been proposed in Chapter 3. It comprises of a small 9-pin stack-able connector for PocketQubes extended to 14 pins for CubeSats. In addition, it employs a single RS-485 data bus, 4 and 8 switched electrical power distribution lines for PocketQubes and CubeSats respectively and an analogue reset line (Section 3.6). The RS-485 data has been selected by performing an extensive trade-off analysis based on the inputs from the CubeSat and PocketQube communities. The Analytical Hierarchy Process (AHP) is used as trade-off method (Section 3.3). It is used to select and weigh the criteria as well as to support qualitative grading of a few criteria. The AHP is executed by distributing a questionnaire to the CubeSat and PocketQube communities leading to 34 and 15 responses respectively. For criteria such as effective data throughput and power consumption, tests have been performed on a representative test setup. The results show that RS-485 provides a significantly higher effective data throughput at significantly lower power consumption in the chosen test setup compared to its competitors I^2C and CAN (Section 3.4). It is however the least implemented data bus in the survey (Section 2.3.1) and has less robustness features compared to CAN. The trade-off is thus sensitive to the weights (Section 3.4.5). RS-485 received the highest priority overall when considering the mean of the community input and at individual level for 56% of the participants for CubeSats and even 86% for PocketQubes. However, the CAN bus receives the highest priority of 32% of the participants for CubeSats. This fraction is expected to grow when its effective data rate would increase and/or power consumption decreases significantly.

For the electrical power distribution, it has been chosen to have a limited set of switched distribution lines to be able to distribute the current per line and be able to detect, isolate and recover from over-current events (Section 3.5). The distribution

is at variable (battery) voltage level to allow the least conversion steps and associated losses. Overall, the lean interface solves the compatibility issues, as identified in existing CubeSats and commercial subsystems, by limiting the options and fixing the pin allocation. Furthermore, the proposed PQ9 interface for PocketQubes and CS14 interface for CubeSats reduces the required board space to just 30% and 8% of PQ60 and PC/104 respectively.

Switching from the commonly used I^2C data bus to RS-485 for housekeeping data provides a 240% increase in effective data rate (Section 3.4.5) and reduces power consumption to 11-42%. Using the electrical power architecture which distributes variable battery voltage to the subsystems saves at least one voltage conversion step (10-15% of the power).

All results on the study on electrical interfaces have contributed to the public release of the PQ9 and CS14 interface standard [1].

Following the research on subsystem interfaces, the research continued on architectural concepts related to the physical arrangement of subsystems and their components. Cellularization in small satellites is a concept in which the functionality of the satellite bus, a subsystem or a component is performed by multiple identical units working in parallel which together fulfill the performance requirements of the satellite. There are technology demonstration missions which have taken this concept to the extreme and applied it to the entire satellite (Section 4.3.1). Mass production of modular units which can meet the different demands for multiple different missions can provide cost benefits through economy of scale. The concept of cellularization also allows for an increase in reliability due to the inherent single component failure tolerance. In contrast to full redundancy, a cellular unit cannot cover the full mission needs by its own. It does, however, allow for graceful degradation when a cellular component fails and the satellite can continue operations at reduced availability or utility. This concept is relatively in-efficient is terms of volume and power consumption at full satellite level compared to non-cellularized concepts. As such, it is not considered to be a feasible concept for all components in a CubeSat or PocketQube. Nevertheless, for a selection of components cellularization can provide specific additional benefits. Cellular reaction wheels (Section 4.4.1) can employ finer control. Cellular 'flat' radios with integrated patch antennas can be used to switch between an omni-directional configuration for a beacon mode and a uni-directional transmission for high-speed data down-link (Section 4.4.4).

A panel concept uses the faces of the satellite to integrate satellite subsystems and components. Also in this case, there is an extreme mission example for which the entire satellite is based on this concept combined with a pyramid shaped internal volume for components on each panel, called 'nano-modular format' (Section 4.3.2). However, this pyramid-shaped volume cannot be used efficiently in practice and can be prohibitive for many payloads that do not fit in the pyramid-shaped envelope. It has been found that the panel concept is suitable to be applied to the integration of components which are relatively flat and require or benefit from placement on the satellite faces. When the associated electronics of those components are subsequently integrated in the panel, such that the wiring harness to the inner subsystems is minimized, this concept saves volume and reduced integration

complexity. A concept of an advanced panel for PocketQubes which integrates the solar cells, the maximum power point tracker, a flat radio, a sun sensor and a global navigation system receiver on a printed circuit board side panel has been proposed (Section 4.4.5).

Finally, the physical breakdown of subsystems in separate modules has been investigated. In a highly modular approach, as commonly applied in CubeSats, each subsystem comprises of one or more separate boards. In this concept, there are for example dedicated boards for the on-board computer, electrical power subsystem and attitude determination and control. While this concept allows for selection of subsystem units based on aspects such as cost, in-flight experience and mission specific needs, it requires many interconnections and similar components to be placed on each board such as micro-controllers and local power regulation. A simple, but effective concept to reduce this apparent overhead is the integration of subsystems, which most CubeSats and PocketQubes have in common, into one single core unit (Section 4.4.6). This can reduce the volume severely as shown by the Delfi-n3Xt 3U CubeSat case study. In a lean configuration, modular non-redundant subsystems and CS14 interfaces are applied. This consumes about half of the volume for the spacecraft bus. In case the advanced panel concept and integrated core unit are applied, the bus volume is reduced to a quarter, leaving the remainder for a payload. Compared to the Delfi-n3Xt launch configuration, which implements redundancy on critical subsystems, the payload volume would increase from 8% to 76% using the proposed advanced concept (Section 4.5).

2.b) Which aspects related to satellite architecture and development mostly affect reliability? How do the options, related to these aspects, compare on reliability using an appropriate metric?

The lean interface proposed in Chapter 3 and the targeted volume reduction using advanced concepts for the satellite bus in Chapter 4 are aimed at CubeSats and PocketQubes with single (non-redundant) subsystems. The proposed concepts increase performance significantly and tackle some of the identified reliability issues associated with interfaces. However, they do not yet address reliability at system level. The implementation of redundancy is a commonly applied method to improve system reliability (Section 5.1.1). This would, however, conflict with the lean philosophy and optimization of payload volume of the proposed concepts. Redundancy is thus considered to be a key reliability aspect in terms of architecture. According to a survey, the average system level testing applied to CubeSats is two months and subsystem redundancy is rarely applied. To improve reliability, additional resources can be allocated to more intensive and complete testing. In terms of development, testing is thus considered to be a key reliability aspect. Given the design choices and available data, it has been decided to focus the reliability analysis for this thesis on the question if CubeSat reliability can best be improved by applying subsystem redundancy or more extensive testing (Section 5.1.1).

The estimated posteriors for subsystems, explained in Section 6.1.1, are used to build a model for CubeSat reliability with subsystem redundancy. For this purpose, a fault tree has been developed. Secondly, models for the failure dependency between the primary and secondary unit of a redundant system have been established

using a failure cause analysis from the survey. Next, a model is developed for the case of improved testing. The model is further extended for the case where the CubeSat is improved iteratively by launching a few demonstration models subsequently and acquiring in-orbit experience. Both cases mitigate immaturity failures compared to the estimates of existing CubeSats from the survey. Using an assessment of this data, a beta distribution model is established for the two cases and used to censor immaturity failure samples from the reference case. To implement the models for the dependencies and immaturity failure mitigation, the fault tree is complemented with a reliability modelling flow (Section 5.7.7). These flows have been used to set up a Monte Carlo simulation for the case of a CubeSat with or without full subsystem redundancy as well as the different improved development cases.

The reliability simulation has shown that subsystem redundancy has a positive impact on satellite reliability for the reference case (Section 5.8). For the case of subsystem redundancy it is assumed that on-board failure detection and a switch to a redundant unit is implemented correctly. This requires significant cost, development time and testing effort. Alternatively, these resources can be allocated to improved testing to reduce immaturity failures instead. The reliability during the useful lifetime (up to 5 years) in this case is better than for a satellite with redundant subsystems. This is true for all simulations up to a period of 5 years, taking into account the uncertainties in the model parameters of subsystems, dependencies and failure mitigation. Beyond a period of 10 years, subsystem redundancy will eventually yield better reliability. This is however only applicable for a few unique missions. Redundancy at subsystem level also consumes twice the volume for the critical subsystems. In the Delfi-n3Xt case study, this leaves less than 10% for the payload, which is prohibitive for some payloads. Finally, the prospects for reliability improvement increase if the satellite bus is iteratively improved for many missions, taking into account the lessons learnt from in-orbit experience (Section 5.9). For example, in the reference case with limited testing, a satellite without redundancy has a reliability of approximately 0.6 after one year while after a few development iterations this increases to 0.95.

6.2. Discussion & Recommendations

In Section 6.1 the answers to the research questions have been presented. In this section, some of the major results are discussed.

The proposed lean electrical interface standard mitigated issues discovered with more versatile interfaces. It results from a trade-off where input from community has been included. The PQ9 and CS14 interface standard is a spin-off product of this thesis. However, it has not been widely adopted outside the TU Delft yet. The process to come to a widely supported standard requires not only taking input from the community, but also intensive discussions with the majority of the stakeholders leading to a compromise which takes the relative stakes and influence of stakeholders in the sector into account. Such a process is outside the scope of this thesis and would potentially have conflicted with the academic approach applied in this thesis. Even in this thesis, the trade-off on data busses has proven to be subjective

to choices in the community as well as the state-of-the-art technology. For example, latest low-power wireless data protocols are rapidly approaching and may even surpass the data rates and power consumption levels of wired interfaces. As long as there is no attempt by a major group of important players in the field to create a worldwide accepted and supported standard electrical interface for PocketQubes and CubeSats, the P9Q and CS14 interfaces are considered to be technically superior over other standards such as PQ60 and PC/104. Also, if an international standardization process takes place, it is highly recommended to at least apply the lean philosophy adopted in this study.

While the question on subsystem redundancy was identified as key consideration for the lean interface and some architectural concepts, the direct impact of these concepts on reliability has not been quantified. For example, the integration of core subsystems into one physical unit can, on one hand, increase reliability by reduction of the total number of components. On the other hand, the lack of a physical breakdown is prohibitive for testing smaller physical units early in the development. Currently, there is not sufficient data available to model both effects appropriately. A comparison between the modular and integrated concept on reliability would require a complete design and test campaign of both concepts using identical human resources. It is questionable if such research is worth the effort since the key consideration for the integration of subsystems is the reduction of volume. The same holds for the advanced outer panel concept. However, for the cellularization concept applied to specific components in which the graceful degradation is considered a key advantage, reliability modelling would be instructive. This requires working out the concept of cellularization in detail first and obtaining sufficient test data which could lead to promising research. Overall, it is recommended to continue the study on a hybrid advanced architecture, explained in Section 6.1.2, by detailed design and prototyping. It should be bread-boarded, iterated and tested to validate the identified advantages compared to state-of-the-art modular architectures. It should also subsequently be demonstrated to convince and mobilize the CubeSat and PocketQube community to adopt the new architectural philosophy. The results may be useful to larger satellites as well.

The satellite reliability assessment in Chapter 5 provides results and methods which can be used beyond the scope of CubeSats and the specific research question. The analysis on the bathtub curve and failure classification (Section 5.2) has led to discarding random failures and introducing a new class of so-called 'immaturity failures' which comprises both infant mortality failures as well as any other single-event failure. It is the antonym of wear-out failures which relate to failures due to accumulative effects. This classification will hold for any satellite size category as well any other complex system which is exposed to a multi-hazard environment. The Lognormal-Gompertz product captures immaturity failure and wear-out failures very well. It provides a higher likelihood than the often-used Weibull-Weibull mixture and solves the unrealistically high lifetimes beyond the observation window. The use of the Kaplan-Meier Estimator, Maximum Likelihood and Bayesian inference are all applied in the analysis and provide a good example of the applicability as well as the advantages and disadvantages of each of these tools. Furthermore,

the satellite model using a reliability activity flow, provides a simulation method and representation which is both sophisticated and easy to interpret. This may be useful for modelling failures of other satellite mass classes or even other complex systems as well. The models used are, however, not perfect. The generalization of failure data from many CubeSats for statistical analysis has shortcomings. Therefore, the generalization comes at the cost of losing qualitative and quantitative information which is mission specific. Contributing factors, such as the maturity of the team, the time spent on testing, the complexity of the satellite design, the ambition of the mission and the design life time, have not been taken into account in the reliability analysis performed. For this study, where modelling has been focused to subsystem redundancy, this has not been considered as a major obstacle. For the development of an individual satellite, however, the discovered parametric models should be updated with mission specific data. Reliability data can be improved through testing, low-level failure modelling and when specifications from the manufacturers are present. As discussed in Section 5.2, the wear-out of subsystems and components relates to deterministic processes (e.g. mechanical wear or cumulative radiation effects) which data can be modelled and/or retrieved early in the design. For immaturity failure, it is recommended to use the general CubeSat parametric models and, when test data becomes available, update the models using Bayesian inference. This reduces the risk that unforeseen failures are ignored or underestimated in reliability analysis.

In terms of reliability, the general conclusion is that CubeSat (as well as PocketQube) developers can best focus on testing and debugging a satellite instead of implementing subsystem redundancy. For specific missions, for example for CubeSats and PocketQubes beyond Earth orbit, which are subject to harsher environments and design lifetimes higher than 10 years, adding subsystem redundancy from early design onward may be justified. The conclusion of this study on subsystem redundancy may not be applicable at component level. Some components have deterministic ageing and wear-out characteristic and the mode (peak of the failure probability density) is in the same order of magnitude as the design life time of the satellite. Typical examples are battery cells and reaction or momentum wheels. For these particular components, cellularization may be an interesting concept to explore. This concept is briefly discussed in Section 4.3.1, but further study with real prototypes and testing is recommended.

6.3. Outlook

With the upcoming launch of Delfi-PQ in 2021, the PQ9 interface and an outer panel with integrated electronics will be demonstrated in-orbit. PocketQubes are also excellent platforms to develop and test the integration of multiple internal subsystems into a single core unit. The launch of CubeSats for the Planet and Spire constellations currently dominate the market in terms of satellites launched. In the next few years, thousands of new satellites will follow as many plans for major CubeSat constellations will become reality. Because the satellite platforms of these constellations are developed iteratively by a major player, the suggested reduction of immaturity failures by series of satellites is already being implemented. An

example of this is the satellites constellation from Planet, which has been developed in iterative cycles adopting agile methods from software industry and implemented several small pre-cursor missions to launch small sets of CubeSats to test the platform and gain experience [2] before launching the bulk of the constellation. PocketQubes, still being in their period of infancy, face a more difficult challenge of acquiring major investments and do not seem to follow the same growth rates yet as CubeSats in the past. For operational missions in the scientific, civil and commercial domains, the utility-over-cost ratio is important as elaborated in a parallel study on PocketQubes [3]. A constellation concept which can attract major investments and/or establishing a consortium of players to develop a very capable and advanced platform in terms of architecture and technology, may accelerate the emergence of PocketQubes. There is however also a threat to the developments of very small satellites. Miniaturization of electronics and electro-mechanical systems will continue for terrestrial applications and subsequently for space systems. However, the size of CubeSats and PocketQubes do pose physical limits on what can be achieved by payloads and the utility of these small satellites can be limited by the available power. Miniaturisation remains a fact, but when specific launch cost is reduced severely it could be used for increasing the performance of larger platforms instead. When this would happen, the number of PocketQubes and CubeSats is eventually expected to decline. Within the CubeSat community, there is already some trend visible of increasing unit size. Several 6U CubeSats have been launched and there are a number of 12U and 16U CubeSats being developed. CubeSats (3U and smaller) and PocketQubes will not completely disappear, but it is expected that on the long term some interest might be lost for missions in which the satellite size is a limiting factor for its utility. However, CubeSats, PocketQubes and even smaller standards, such as a satellite-on-a-chip [4], will likely remain for education and technology demonstration objectives because a reduction of launch cost and further miniaturization will continue to expand the access to space for (new) players with limited resources and high ambitions. These small satellite platforms also provide excellent opportunities to demonstrate advanced distributed space system concepts such as formation flying and fractionated spacecraft and can be used to demonstrate space debris mitigation techniques. Also for deep space missions, for which launch capacity is smaller and cost is higher, very small satellites are expected to gain interest.

References

[1] J. Bouwmeester, *PQ9 and CS14 Electrical and Mechanical Subsystem Interface Standard for PocketQubes and CubeSats*, Tech. Rep. (Delft University of Technology, 2018).

[2] C. R. Boshuizen, J. Mason, P. Klupar, and S. Spanhake, *Results from the Planet Labs Flock Constellation,* in *Proceedings of the 28th AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2014).

[3] J. Bouwmeester, S. Radu, M. S. Uludag, N. Chronas, S. Speretta, A. Menicucci, and E. K. Gill, *Utility and Constraints of PocketQubes,* CEAS Space Journal **12**, 573 (2020).

[4] D. J. Barnhart, T. Vladimirova, and M. N. Sweeting, *Very-small-satellite Design for Distributed Space Missions,* Journal of Spacecraft and Rockets **44** (2007), 10.2514/1.28678.

Curriculum Vitæ

Jasper Bouwmeester

25-04-1981	Born in 's-Gravenzande, the Netherlands.

Education

1993–1999	Atheneum Alkwin Kollege, Uithoorn
1999–2005	BSc. in Aerospace Engineering Delft University of Technology
2005–2007	MSc. in Aerospace Engineering Delft University of Technology
2014–2021	PhD. in Aerospace Engineering Delft University of Technology

Experience

2007–	Lecturing in several courses and supervising MSc. thesis studies
2007–	Researcher small satellite technology
2013–	Coordinator of BSc minor curriculum on spaceflight
2007–2018	Delfi Satellite Program Manager
2008–2018	Delfi-C^3 Project Manager
2007–2014	Delfi-n3Xt Project Manager
2016–2018	Delfi-PQ Project Manager

List of Publications

Publications [1–4] directly relate to this thesis. Publications [5–15] are written as main author and [16–48] as co-author outside the scope of this thesis.

[1] J. Bouwmeester, E. Gill, S. Speretta, and S. Uludag, *A New Approach on the Physical Architecture of CubeSats & PocketQubes,* Journal of the Britisch Interplanetary Society **71**, 1 (2018).

[2] J. Bouwmeester, S. P. van der Linden, A. Povalac, and E. K. Gill, *Towards an innovative electrical interface standard for PocketQubes and CubeSats,* Advances in Space Research **62**, 3423 (2018).

[3] J. Bouwmeester, M. Langer, and E. Gill, *Survey on the Implementation and Reliability of CubeSat Electrical Bus Interfaces,* CEAS Space Journal **9** (2017), 10.1007/s12567-016-0138-0.

[4] J. Bouwmeester and N. Santos, *Analysis of the Distribution of the Electrical Power in CubeSats,* in *Proceedings of the 4S symposium* (ESA, Valetta, 2014).

[5] J. Bouwmeester, S. Radu, M. S. Uludag, N. Chronas, S. Speretta, A. Menicucci, and E. K. Gill, *Utility and Constraints of PocketQubes,* CEAS Space Journal **12**, 573 (2020).

[6] J. Bouwmeester, L. Rotthier, C. Schuurbiers, W. T. Wieling, G. V. D. Horn, F. Stelwagen, E. Timmer, and M. Tijssen, *Preliminary Results of the Delfi-n3Xt Mission,* in *Proceedings of the 4S Symposium* (ESA, Mallorca, 2014).

[7] J. Bouwmeester, C. A. H. Schuurbiers, F. Stelwagen, M. Tijssen, and E. Timmer, *Delfi-n3Xt : A Powerful Triple-unit CubeSat for Education and Technology Demonstration,* in *Proceedings of the 2nd IAA Conference on University Satellite Mission and CubeSat Workshop* (IAA, Rome, 2013).

[8] J. Bouwmeester, J. Reijneveld, T. Hoevenaars, and D. Choukroun, *Design and Verification of a Very Compact and Versatile Attitude Determination and Control System for the Delfi-n3Xt Nanosatellite,* in *Proceedings of the 4S Symposium* (ESA, Portoroz, 2012).

[9] J. Bouwmeester, E. Gill, P. Sundaramoorthy, and H. Kuiper, *Concept Study of a LEO Constellation of Nanosatellites for Near Real Time Optical Remote Sensing,* in *Proceedings of the 62nd International Astronautical Congress*, edited by IAF (Cape Town, 2011).

[10] J. Bouwmeester and J. Guo, *Survey of Worldwide Pico- and Nanosatellite Missions, Distributions and Subsystem Technology,* Acta Astronautica **67**, 854 (2010).

[11] J. Bouwmeester, G. F. Brouwer, E. K. Gill, G. L. Monna, and J. Rotteveel, *Design Status of the Delfi-Next Nanosatellite Project,* in *Proceedings of the 61st International Astronautical Congress* (IAF, Prague, 2010).

[12] J. Bouwmeester and R. Hamann, *Delfi-n3Xt Nanosatellite Subsystems: Buying, Outsourcing or Internal Development,* in *Small Satellites for Earth Observation: 7th International Symposium of the International Academy of Astronautics (IAA)* (Berlin, 2009).

[13] J. Bouwmeester, G. Aalbers, and W. Ubbels, *Preliminary Mission Results and Project Evaluation of the Delfi-C3 Nano-Satellite,* in *Proceedings of the 4S Symposium* (ESA, Rhodes, 2008).

[14] J. Bouwmeester, E. K. A. Gill, and C. J. M. Verhoeven, *Advancing Nano-Satellite Platforms : the Delfi Program,* in *Proceedings of the 59th International Astronautical Congress* (IAF, Glasgow, 2008).

[15] J. Bouwmeester, R. Amini, and R. Hamann, *Command and Data Handling Subsystem for a Satellite Without Energy Storage: Delfi-C3,* in *Proceedings of the 58th International Astronautical Congress* (IAF, Hyderabad, 2007).

[16] M. S. Uludag, E. Yakut, S. Speretta, S. Radu, N. C. Foteinakis, J. Bouwmeester, A. Menicucci, and E. Gill, *Getting the Delfi-PQ Ready for Multiple Launch Options,* in *Proceedings of the International Symposium on Space Technology and Science* (Fukui, 2019).

[17] S. Speretta, M. S. Uludag, V. Karunanithi, S. Radu, N. C. Foteinakis, J. Bouwmeester, A. Menicucci, and E. Gill, *A Multi Frequency Deployable Antenna System for Delfi-PQ,* in *Proceedings of the International Symposium on Space Technology and Science* (Fukui, 2019).

[18] S. Radu, M. S. Uludag, S. Speretta, J. Bouwmeester, E. Gill, and N. Foteinakis, *Delfi-PQ: the First PocketQube of Delft University of Technology,* in *Proceedings of the 69th International Astronautical Congress* (IAF, Bremen, 2018).

[19] M. Uludag, S. Speretta, J. Bouwmeester, E. Gill, and T. PerezSoriano, *A New Electrical Power System Architecture for Delfi-PQ,* in *Proceedings of the 4th IAA Conference on University Satellite Missions and Cubesat Workshop* (Rome, 2017).

[20] M. Langer, M. Weisgerber, J. Bouwmeester, and A. Hoehn, *A reliability estimation tool for reducing infant mortality in Cubesat missions,* in *IEEE Aerospace Conference Proceedings* (2017).

[21] T. Vergoossen, J. Guo, J. Bouwmeester, and W. A. Groen, *Design, Integration, and Testing of World's Smallest Satellite Reaction Wheel,* in *Proceedings of the International Astronautical Congress, IAC* (2017).

[22] J. Llanos and J. Bouwmeester, *Thermoelectric Harvesting for an Autonomous Self-Powered Temperature Sensor in Small Satellites,* in *Proceedings of the 68th International Astronautical Congress* (IAF, Adelaide, 2017).

[23] S. van der Linden, J. Bouwmeester, and A. Povalac, *Design and Validation of an Innovative Data Bus Architecture for CubeSats,* in *14th Reinventing Space Conference* (2016).

[24] K. Cowan, A. Cervone, J. Bouwmeester, D. Stam, E. Schrama, R. Hanssen, and C. Verhoeven, *Multidisciplinary space education in a blended learning environment: The new Spaceflight minor at Delft university of technology,* in *Proceedings of the International Astronautical Congress, IAC* (2016).

[25] J. Guo, J. Bouwmeester, and E. Gill, *In-orbit Results of Delfi-n3Xt: Lessons Learned and Move Forward,* Acta Astronautica **121**, 39 (2016).

[26] S. Speretta, T. Soriano, J. Bouwmeester, J. Carvajal-Godinez, A. Menicucci, T. Watts, P. Sundaramoorthy, J. Guo, and E. Gill, *CubeSats to Pocketqubes: Opportunities and challenges,* in *Proceedings of the International Astronautical Congress* (IAF, Guadalajara, 2016).

[27] M. Langer and J. Boumeester, *Reliability of CubeSats – Statistical Data, Developers' Beliefs and the Way Forward,* in *Proceedings of 30th AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2016).

[28] R. Schoemaker and J. Bouwmeester, *Evaluation of Bluetooth Low Energy Wireless Internal Data Communication for Nanosatellites,* in *Proceedings of the 4S symposium* (ESA, Valetta, 2014).

[29] J. Guo, J. Bouwmeester, and E. Gill, *In-orbit results of Delfi-n3Xt: Lessons learned and move forward,* in *Proceedings of the International Astronautical Congress, IAC*, Vol. 6 (2014).

[30] A. Telgie, J. Bouwmeester, and F. Stelwagen, *Flight Results and Lessons Learnt From the Delfi-n3Xt Electrical Power Subsystem Operations,* in *Proceedings of the International Astronautical Congress*, Vol. 5 (IAF, Toronto, 2014).

[31] J. Guo, J. Bouwmeester, and E. Gill, *From Single to Formation Flying CubeSats: An Update of the Delfi Programme,* in *Proceedings of the AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2013).

[32] E. Gill, P. Sundaramoorthy, J. Bouwmeester, B. Zandbergen, and R. Reinhard, *Formation Flying Within a Constellation of Nano-satellites: The QB50 Mission,* Acta Astronautica **82** (2013), 10.1016/j.actaastro.2012.04.029.

[33] M. Vázquez, J. Bouwmeester, and O. López, *Improving communicationi link budgets for small satellites,* in *Proceedings of the International Astronautical Congress, IAC*, Vol. 10 (2013).

[34] T. Hoevenaars, S. Engelen, and J. Bouwmeester, *Model-based discrete PID controller for cubesat reaction wheels based on cots brushless DC motors,* in *Advances in the Astronautical Sciences*, Vol. 145 (2012) pp. 379–394.

[35] J. Kuiper and J. Bouwmeester, *A 1.5U Cube-Sat Camera Cornerstone Design for a Multiple Aperture Earth Observation System,* in *Proceedings ot the 62nd International Astronautical Congress*, Vol. 3 (IAF, Cape Town, 2011).

[36] M. Raif, U. Walter, and J. Bouwmeester, *Dynamic System Simulation of Small Satellite Projects,* Acta Astronautica **67**, 1138 (2010).

[37] T. Jansen, A. Reinders, G. Oomen, and J. Bouwmeester, *Performance of the First Flight Experiment with Dedicated Space CIGS cells Onboard the Delfi-C3 Nanosatellite,* in *Proceedings of the IEEE Photovoltaic Specialists Conference* (IEEE, Honolulu, 2010) pp. 1128–1133.

[38] E. Dekens, G. Brouwer, J. Bouwmeester, and J. Kuiper, *Development of a Nano-Satellite Reaction Wheel System with Commercial Off-The-Shelf Motors,* in *Proc. 7th ESA Round Table for Space Applications* (Noordwijk, 2010).

[39] E. Gill, P. Sundaramoorthy, J. Bouwmeester, B. Sanders, and C. Science, *Formation Flying to Enhance the QB50 Space Network,* in *Proceedings of 4S Symposium* (ESA, Funchal, 2010).

[40] P. Sundaramoorthy, E. Gill, C. Verhoeven, and J. Bouwmeester, *Two CubeSats with Micro-Propulsion in the QB50 Satellite Network,* in *Proceedings of the 24th AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2010) pp. 1–11.

[41] G. F. Brouwer and J. Bouwmeester, *From the Delfi-C3 Nano-satellite towards the Delfi-n3Xt Nano-satellite,* in *Proceedings of the 23rd AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2009) pp. 1–9.

[42] L. Lebbink, R. Hamann, J. Bouwmeester, and G. Brouwer, *Interface control procedures for university satellite pro-grammes,* in *60th International Astronautical Congress 2009, IAC 2009*, Vol. 5 (2009).

[43] N. Cornejo, J. Bouwmeester, and G. Gaydadjiev, *Implementation of a Reliable Data Bus for the Delfi Nanosatellite Programme,* in *Proceedings of the 7th IAA Symposium on Small Satellites for Earth Observation* (IAA, Berlin, 2009).

[44] M. Genbrugge, R. Teuling, J. Kuiper, G. Brouwer, and J. Bouwmeester, *Configuration management in nanosatellites projects: Evaluation of Delfi-C3 and consequent adaptation for Delfi-n3Xt,* in *Proceedings of the 60th International Astronautical Congress* (IAF, Daejon, 2009).

[45] R. Hamann, J. Bouwmeester, and G. Brouwer, *Delfi-C3 Preliminary Mission Results,* in *Proceedings of the 23rd AIAA/USU Small Satellite Conference* (AIAA, Logan, 2009).

[46] S. D. Jong, G. T. Aalbers, and J. Bouwmeester, *Improved Command and Data Handling System for the Delfi-n3Xt Nanosatellite,* in *Proceedings of the 59th International Astronautical Congress* (IAF, Glasgow, 2008).

[47] W. J. Ubbels, C. J. M. Verhoeven, R. J. Hamann, E. Gill, and J. Bouwmeester, *First flight results of the Delfi-C3 satellite mission,* in *Proceedings of the 22nd AIAA/USU Conference on Small Satellites* (AIAA, Logan, 2008).

[48] S. Y. Go, J. Bouwmeester, and G. F. Brouwer, *Optimized Three-unit CubeSat Structure for Delfi-n3Xt,* in *Proceedings of the 59th International Astronautical Congress* (IAF, Glasgow, 2008).