Evolving Systems Approach to the Attitude Control of a Space-Debris Removal Spacecraft

Habets, Jannick; Mooij, E.; Frost, Susan A.

DOI
10.2514/6.2017-1039

Publication date
2017

Document Version
Accepted author manuscript

Published in
AIAA Guidance, Navigation, and Control Conference, 2017

This is an Accepted Manuscript of an article published in: AIAA Guidance, Navigation, and Control Conference
January 9-13, Grapevine, TX. ISBN: 978-1-62410-450-3
Available online: https://arc.aiaa.org/doi/10.2514/6.2017-1039

Evolving Systems Approach to the Attitude Control of a Space-Debris Removal Spacecraft

J.M.G. Habets* and E. Mooij†
Delft University of Technology, Faculty of Aerospace Engineering,
Kluyverweg 1, 2629 HS Delft, The Netherlands

S.A. Frost‡
NASA Ames Research Center, Moffett Field, CA 94035

The unexpected loss of contact with Envisat, the European Space Agency's largest environmental spacecraft, has added one more item to the growing list of debris in space. ESA plans to retrieve Envisat by attaching a chaser spacecraft to it with the help of a robot arm and tentacles. This paper investigates the stability and controllability of the system before, during, and after docking operations. A new approach, called Evolving Systems, is applied to the problem by modelling the connection between Envisat and the chaser as an increasingly stiff (rotational) spring-damper system. The stability of the system is assessed by a linear stability analysis and a nonlinear system analysis. For the nonlinear analysis, a simulator is used consisting of the two spacecraft, the connection model, a Linear Quadratic Regulator, a reaction-control system, and a control allocator. The linear stability analysis showed that the system remains stable during its evolution; however, if the debris would have the same size as the chaser, instability can occur. Further research could focus on docking with smaller space debris. The nonlinear analysis showed that a simple Linear Quadratic Regulator can stabilise the system before, during, and after docking, even if the initial conditions of the target are uncertain. Moreover, the docking phase has little effect on the motion of Envisat and the stability of the system, because of the large difference in size between the chaser and Envisat.

Nomenclature

$\mathbf{a}$	Actuator control input vector
$\mathbf{e}$	Euler eigenaxis
$\mathbf{I}$	Inertia tensor, kg·m^2
J	Performance index
$\mathbf{K}$	Gain matrix
m	Mass, kg
$\mathbf{M}$	External torque, Nm
$\mathbf{q}$	Quaternion
$\mathbf{Q}$	State weight matrix
$\mathbf{R}$	Control input weight matrix
$\mathbf{T}$	Configuration matrix
$\mathbf{x}$	State vector
$\mathbf{u}$	Control vector
$\mathbf{y}$	Output vector
ϵ	Connection parameter

*MSc Graduate, section Astrodynamics and Space Missions.
†Assistant Professor, section Astrodynamics and Space Missions, e.mooij@tudelft.nl, Associate Fellow AIAA.
‡Research Engineer, Intelligent Systems Division, POB 1, M/S 269-3, Member AIAA

θ	Rotation angle around Euler eigenaxis, rad
λ	Weight factor control allocation
$\boldsymbol{\omega}$	Angular velocity vector, rad/s

Subscripts

0	Initial condition
C	Chaser
e	Error
S	Stack
T	Target

I. Introduction

A diverse group of objects contribute to space debris; large objects, such as non-functional satellites, spent upper-stages of rockets, payload adapters, lens covers, clamp-bands, yo-yo de-spin devices, and even screwdrivers and protective gloves lost during extra-vehicular activities of astronauts. But also smaller objects, such as slag particles produced during solid rocket motor burns, cooling liquids, and degradation products resulting from crack formation.[1]

All this debris is becoming an increasing threat for future space missions. Therefore, numerous options to actively get rid of space debris are being evaluated. They range from solar sails to sling shots, and from tentacles to tethers. Some are more suited for small debris and some for large debris. The European Space Agency (ESA) aims to actively remove debris with the e.Deorbit mission, which is part of the Clean Space Initiative.

e.Deorbit is an active debris removal (ADR) mission, designed for debris with a heavy mass in Sun-synchronous/polar orbits, between 800 km and 1000 km altitude. Concurrent Design Facility (CDF) studies have been performed with assumptions about the target based on Envisat, ESA's largest environmental satellite, which became inoperable in April 2012 and is a candidate for a deorbit mission. Amongst several options to actively remove Envisat from its current orbit there is one, which uses a *chaser* satellite with a robotic arm.[2] The chaser satellite will attach a robotic arm to the target, fix itself to the target with tentacles, stabilise both vehicles, and eventually deorbit.

The focus of this research is on the attitude control of the chaser when it is in the vicinity of the target. The stability of the system is evaluated before, during, and after docking with the large space debris. To facilitate the stability analysis, a new method called *Evolving Systems* is used. Evolving Systems are dynamical systems that are self-assembled from actively controlled subsystems.[3] The docking of the chaser with Envisat can be seen as such a system, except that the debris is passive. As mentioned before, docking is a complex procedure, consisting of using a robot arm, tentacles, and pushing rods. To simplify the modelling of the docking procedure, Evolving Systems can be used to represent the forces and moments acting on the various components by springs and dampers, which become stiffer as the docking progresses.

This paper will start by detailing the docking scenario and the dynamics of the system. Next, Evolving Systems will be introduced, together with a mathematical background and an analysis of the chaser-target system. Then, the elements of the attitude control system of the chaser will be discussed. This includes a Linear Quadratic Regulator (LQR), a control allocation algorithm, and a thruster model. Next, the simulation results will be presented and evaluated. This paper ends with conclusions and recommendations.

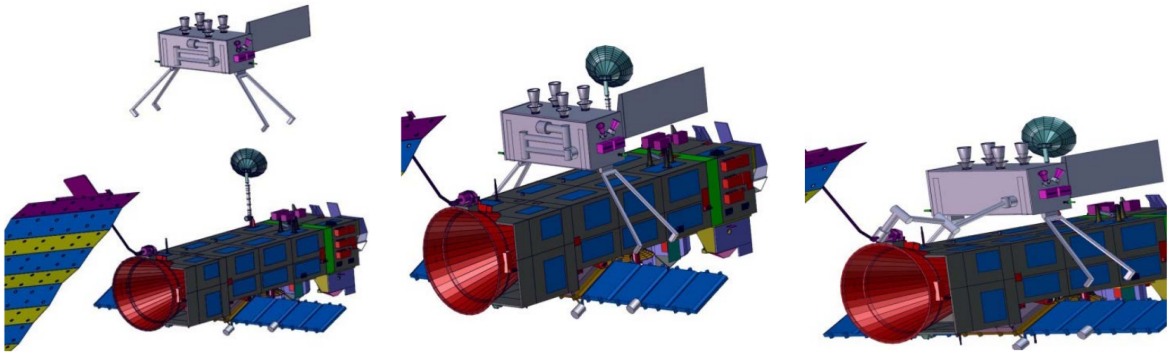
II. Docking Model

A. Docking Scenario

The mission scenario starts with a launch and commissioning phase. After orbit transfer and phasing, the rendezvous phase will start at a relative distance of around 3 km. When the chaser has approached the target to within 50 m, the target-observation phase commences. Then, the chaser begins to synchronise its motion to that of the target and will eventually capture it.

A proposed rendezvous-and-docking procedure can be seen in Fig. 1. When the chaser is in the vicinity of Envisat, it will deploy its tentacles (Fig. 1a) and match the motion of the target. Because of a possible, large

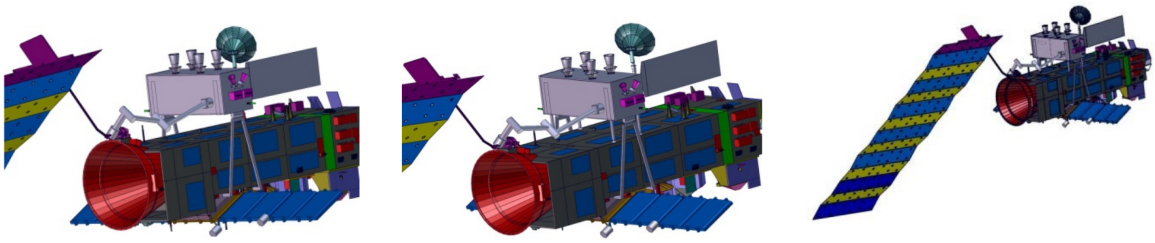
tumbling motion of Envisat of up to 3.5 deg/s (Ref. 2), the chaser has to actively control its translational motion to stay in the same position with respect to Envisat. Furthermore, to keep the chaser pointed towards the target, the chaser has to control its attitude. Meanwhile, during the *synchronisation*, the chaser will approach Envisat on a certain path towards the contact point (Fig. 1b). When the chaser is close enough, it will deploy its robotic arm and grab Envisat's solar-panel beam (Fig. 1c). With this point of support between the chaser and the target, the target has better manoeuvrability and can position itself closer towards the intended contact point. When the chaser has moved to its final position, it will close its tentacle arms around the body of Envisat (Fig. 1d) and deploy the pushing rods to firmly attach itself to Envisat (Fig. 1e). These three phases have been called the *semi-connected* phase. The next task of the chaser is to stabilise the rotational motion of the *stack* (Fig. 1f) and possibly re-orient itself to attain a sun-pointing attitude or braking attitude required for the deorbit burns.



(a) Deploy the tentacles

(b) Approach Envisat

(c) Deploy robotic arm and grab Envisat solar panel beam



(d) Close the tentacle arms

(e) Deploy the pushing rods

(f) Stabilise stack for deorbiting

Figure 1: Docking phases[2]

B. System Dynamics

The attitude of both the target and chaser spacecraft are parametrised by quaternions of rotation. The four-dimensional quaternion $\mathbf{q} = [q_1 \; q_2 \; q_3 \; q_4]^T$ is described as:

$$\begin{aligned} q_1 &= e_x \sin(\theta/2) \\ q_2 &= e_y \sin(\theta/2) \\ q_3 &= e_z \sin(\theta/2) \\ q_4 &= \cos(\theta/2) \end{aligned} \tag{1}$$

where θ is the rotation angle around the Euler eigenaxis $\mathbf{e} = [e_x\ e_y\ e_z]^T$. The kinematic differential equations in terms of quaternions are:

$$\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} = \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{2}$$

where $\boldsymbol{\omega}_B^{B/I} = [\omega_x\ \omega_y\ \omega_z]^T$ is the angular velocity of the vehicle with respect to the ECI frame, expressed in the vehicle body frame, which will be simply denoted as $\boldsymbol{\omega}$ from here on.

The rotational dynamics of a rigid body are described with Euler's equations of rotational motion:

$$\mathbf{M} = \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} \tag{3}$$

with $\mathbf{M}$ the external torque, $\mathbf{I}$ the inertia tensor of the body, and $\boldsymbol{\omega}$ the angular-velocity vector.

The moments of inertia of Envisat around the principle axes are equal to[a]

$$\mathbf{I}_T = \begin{bmatrix} 16969 & 0 & 0 \\ 0 & 124700 & 0 \\ 0 & 0 & 129077 \end{bmatrix} \text{ kg m}^2 \tag{4}$$

The moment of inertia of the chaser is based on a rectangular box with dimensions $1.2 \times 1.2 \times 3$ m, and a mass of 1,500 kg (Ref. 2):

$$\mathbf{I}_C = \begin{bmatrix} \frac{1}{12}m\left(y^2+z^2\right) & 0 & 0 \\ 0 & \frac{1}{12}m\left(x^2+z^2\right) & 0 \\ 0 & 0 & \frac{1}{12}m\left(x^2+y^2\right) \end{bmatrix} = \begin{bmatrix} 1320 & 0 & 0 \\ 0 & 1320 & 0 \\ 0 & 0 & 360 \end{bmatrix} \text{ kg m}^2 \tag{5}$$

When the chaser has docked with the target, it is assumed that a rigid body is formed and that there is no flexibility between the chaser and target. This stack configuration has the combined inertia of both the chaser and the target. First, the centre of mass of the combined system has to be found

$$\mathbf{r}_{cm} = \frac{1}{M}\left(m_T\mathbf{r}_T + m_C\mathbf{r}_C\right) \tag{6}$$

where $M = m_T + m_C$. It is assumed that chaser docks exactly in line with the centre of mass of Envisat.

Next, using the parallel-axis theorem, the inertias of the chaser and target can be found with respect to the new centre of mass of the stack:

$$\mathbf{I}_S^C = \mathbf{I}_C^C - \mathbf{M}^C[\mathbf{r}_{cm}^C\times]^2, \qquad \mathbf{I}_S^T = \mathbf{I}_T^T - \mathbf{M}^T[\mathbf{r}_{cm}^T\times]^2 \tag{7}$$

with

$$\mathbf{M} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix}, \qquad [\mathbf{r}_{cm}\times] = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{8}$$

Adding the two gives the inertia of the stack

$$\mathbf{I}_S = \mathbf{I}_S^T + \mathbf{I}_S^C = \begin{bmatrix} 130521 & 0 & -30 \\ 0 & 27282 & 0 \\ -30 & 0 & 134251 \end{bmatrix} \text{ kg m}^2 \tag{9}$$

[a]ESA, "Envisat FINAL MARB ESTEC", 2012, ESA Document

III. Evolving Systems

In Ref. 3 a framework is proposed for the autonomous assembly of actively controlled dynamical subsystems into an Evolved System. Originally, Evolving Systems were intended for the assembly of large space structures, such as large telescopes, space stations, or large solar-collecting spacecraft. However, Evolving Systems can also be used for rendezvous and docking missions, servicing missions, or formation flying.

An Evolving System consists of a number of actively controlled subsystem components. These components assemble, or mate, to form an Evolved System, which has a higher purpose than the individual subsystem components. For example, a chaser satellite approaches some large space debris and connects with it to form a system that will deorbit itself. It is important to note that in this particular system only one of the components, i.e., the chaser, is actively controlled.

The connections between components in the Evolving System are modelled as compliant forces operating on the displacements of physical coordinates (translational or rotational) within the components. The connection forces joining the components can, e.g., be modelled as springs connecting two elements. A key concept in Evolving Systems is an evolutionary connection parameter ϵ ranging from 0 to 1. This parameter allows the connection between the two components to evolve smoothly from unconnected ($\epsilon = 0$) to connected ($\epsilon = 1$).

In Evolving Systems, it is assumed that the control of the components is autonomous. The individual components are controlled locally, to remain stable and to meet their performance requirements. An important issue in Evolving Systems is the stability of the components during the evolution of the system. Stability cannot always be ensured, using only local controllers. Therefore, an adaptive controller is placed on a key component to restore stability of the evolving system in Ref. 4. One of the goals of this paper is to determine if a simple Linear Quadratic Regulator can keep the system stable as well, or if a more advanced adaptive controller is required.

In this section a mathematical description of Evolving Systems will be presented. Next, a simple spring-damper model, which is used to describe the chaser-target system, will be treated. Last, a linear stability analysis will be performed.

A. Mathematical background

Starting with a general nonlinear state equation for a system of i components:[3]

$$\left.\begin{aligned} \dot{\mathbf{x}}_i &= f_i(\mathbf{x}_i, \mathbf{u}_i); \quad \mathbf{x}_i(0) = \mathbf{x}_i^0 \\ \mathbf{y}_i &= g_i(\mathbf{x}_i, \mathbf{u}_i) \end{aligned}\right\} \qquad i = 1, 2, ..., L \tag{10}$$

where, $\mathbf{x}_i = \left[x_1^i, x_2^i, ..., x_{n_i}^i\right]^T$ is the component state vector with n_i denoting the length of the i^{th} component, $\mathbf{u} = \left[u_1^i, u_2^i, ..., u_{m_i}^i\right]^T$ the control input vector (dimensions m_i), and $\mathbf{y}_i = \left[y_1^i, y_2^i, ..., x_{p_i}^i\right]^T$ the output vector (dimensions p_i). Both f_i and g_i can be nonlinear functions. In the original formulation of Evolving Systems, each component is locally controlled, meaning that the control depends only on the local state or local output, i.e. $\mathbf{u}_i = h_i(\mathbf{x}_i)$ or $\mathbf{u}_i = h_i(\mathbf{y}_i)$. However, in this research the components are not restricted to be locally controlled; knowledge of the state of the debris is required to synchronise the motion of the chaser with the debris.

The idea behind Evolving Systems is to add connection forces and moments between various components. A connection between component i and j is described by $k_{ij}(\mathbf{x}, \mathbf{u})$ and the evolution of the connection is captured in the parameter ϵ_{ij}, which ranges from 0 to 1:

$$\left.\begin{aligned} \dot{\mathbf{x}}_i &= f_i(\mathbf{x}_i, \mathbf{u}_i) + \sum_{j=1}^{L} \epsilon_{ij} k_{ij}(\mathbf{x}, \mathbf{u}); \quad \mathbf{x}_i(0) = \mathbf{x}_i^0 \\ \mathbf{y}_i &= g_i(\mathbf{x}_i, \mathbf{u}_i) \end{aligned}\right\} \qquad i = 1, 2, ..., L \tag{11}$$

In this paper, the connection is modelled as a simple spring-damper system. So the dynamics of the complex robot arm, tentacles arms, and pushing rods are assumed to be captured by a simple (rotational) spring-damper system. In the following, a parametrisation of the spring and damper will be given.

B. Spring parametrisation

Euler's eigenaxis can be used as the single axis around which the spring is rotated from its equilibrium position. Figure 2 shows two reference frames A and B. By rotating frame A around the eigenaxis $\mathbf{e}$ by an angle θ frame B is reached.

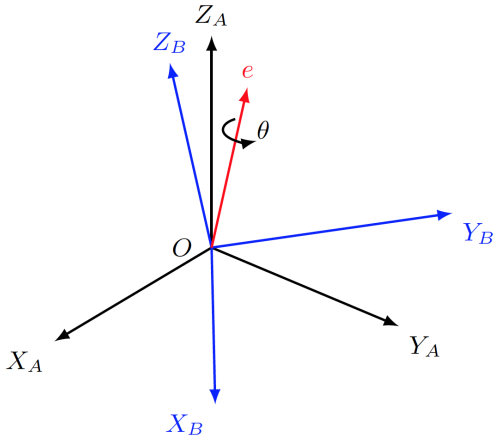


Figure 2: Euler's eigenaxis

Imagining that the spring is connected to the Euler eigenaxis, we can write the spring moment as:

$$\mathbf{M} = -\mathbf{K}\theta\mathbf{e} \tag{12}$$

where it is assumed that the spring-constant matrix $\mathbf{K}$ consists only of diagonal terms. The spring moment can then be written as:

$$\mathbf{M} = \begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} = \begin{pmatrix} -e_x k_x \theta \\ -e_y k_y \theta \\ -e_z k_z \theta \end{pmatrix} \tag{13}$$

with k_x, k_y, and k_z the spring constants around the three axes. So, for example, the single rotation around the Z_B axis gives an eigenvector $\mathbf{e} = [0\ 0\ 1]^T$.

To apply this to the rotations of the target and chaser, the inertial frame I will be used as the reference frame to which the rotations of the chaser and target will be described. In terms of direction cosine matrix, the rotation from the target to the chaser frame can be described as

$$\begin{aligned} \mathbf{C}^{C/T} &= \mathbf{C}^{C/I}\mathbf{C}^{I/T} \\ &= \mathbf{C}^{C/I}\left(\mathbf{C}^{T/I}\right)^T \end{aligned} \tag{14}$$

and from the chaser to the target

$$\begin{aligned} \mathbf{C}^{T/C} &= \mathbf{C}^{T/I}\mathbf{C}^{I/C} \\ &= \mathbf{C}^{T/I}\left(\mathbf{C}^{C/I}\right)^T \\ &= \left(\mathbf{C}^{C/T}\right)^T \end{aligned} \tag{15}$$

Or, in terms of quaternions:

$$\begin{aligned} \mathbf{q}^{C/T} &= \mathbf{q}^{C/I} \otimes \left(\mathbf{q}^{T/I}\right)^{-1} \\ \mathbf{q}^{T/C} &= \mathbf{q}^{T/I} \otimes \left(\mathbf{q}^{C/I}\right)^{-1} = \left(\mathbf{q}^{C/T}\right)^* \end{aligned} \tag{16}$$

Then, the spring moment acting on, e.g., the chaser can be calculated using the quaternion from the T frame to the C frame, $\mathbf{q}^{C/T}$:

$$\mathbf{M}_C = \begin{pmatrix} -e_x^{C/T} k_x \theta \\ -e_y^{C/T} k_y \theta \\ -e_z^{C/T} k_z \theta \end{pmatrix} \tag{17}$$

The spring moment on the target is the negative of this after recognising that the eigenaxis is constant in both reference frames:

$$\begin{aligned}\mathbf{e} &= e_1\hat{\mathbf{x}}_C + e_2\hat{\mathbf{y}}_C + e_3\hat{\mathbf{z}}_C \\ &= e_1\hat{\mathbf{x}}_T + e_2\hat{\mathbf{y}}_T + e_3\hat{\mathbf{z}}_T\end{aligned} \tag{18}$$

C. Damping

The damping moment $\mathbf{M}_D$ of a simple, mechanical viscous rotational damper (dashpot) is given by

$$\mathbf{M}_D = -\mathbf{C}_d\boldsymbol{\omega}_e \tag{19}$$

where $\mathbf{C}_d$ is the damping coefficient matrix, and the error angular-velocity vector $\boldsymbol{\omega}_e$, as seen from, e.g., the target, is equal to the angular velocity of the chaser minus that of the target. It can be expressed in the target frame, as:

$$\boldsymbol{\omega}_T^e = \boldsymbol{\omega}_T^{C/T} = \boldsymbol{\omega}_T^{C/I} + \boldsymbol{\omega}_T^{I/T} \tag{20}$$

$$= \boldsymbol{\omega}_T^{C/I} - \boldsymbol{\omega}_T^{T/I} \tag{21}$$

$$= \mathbf{T}^{T/C}\boldsymbol{\omega}_C^{C/I} - \boldsymbol{\omega}_T^{T/I} \tag{22}$$

where $\boldsymbol{\omega}_T^{I/T} = -\boldsymbol{\omega}_T^{T/I}$ has been used, and the chaser angular velocity has been expressed in the target reference frame.

Similarly, for the chaser

$$\boldsymbol{\omega}_C^e = \boldsymbol{\omega}_C^{T/C} = \mathbf{T}^{C/T}\boldsymbol{\omega}_T^{T/I} - \boldsymbol{\omega}_C^{C/I} \tag{23}$$

which is equal to minus the error angular velocity seen from the target.

So, if we want the damper moment acting on the target, $\mathbf{M}_D^T$, we use the error angular velocity as seen from the target, $\boldsymbol{\omega}_T^{C/T}$:

$$\mathbf{M}_D^T = -\mathbf{C}_d\boldsymbol{\omega}_T^{C/T} \tag{24}$$

D. Linear stability

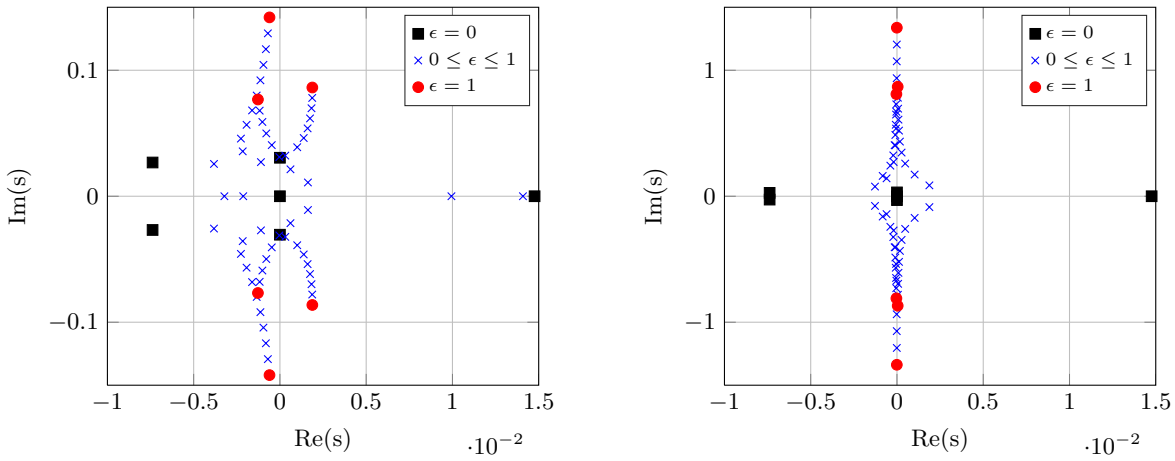
The linear stability of the chaser-target system can be analysed by looking at a linearised version of the system. To simplify the analysis it is assumed that, instead of the chaser being attached to a moving target, it is attached to a stationary wall. The reasoning behind this is that the rotational dynamics of the chaser have very little impact on the target debris because of the large (two orders of magnitude) inertia difference between the chaser and target.

Following the method in Ref. 3, the stability of the linear system is assessed by looking at the location of the poles during the evolution from unconnected to connected. Figure 3 shows the location of the poles for an uncontrolled system. It can be seen that for a system with a small spring constant some of the poles remain in or move into the right half-plane, meaning the system becomes unstable. For a large spring constant, the system remains stable by itself.

The same analysis was performed but with an Linear Quadratic Regulator as a local controller on the chaser, as explained in Sec. IV.A. Figure 4 shows the locations of the poles for a system with an LQR with both high and low control gains. As can be seen, the system remains stable during its evolution. Increasing the gains of the LQR moves the poles further along both the imaginary and real axis, meaning that the system damps out faster. Based on this linear stability analysis it can be concluded that for a controlled system, where the target is much larger than the chaser, the system does not become unstable as it evolves.

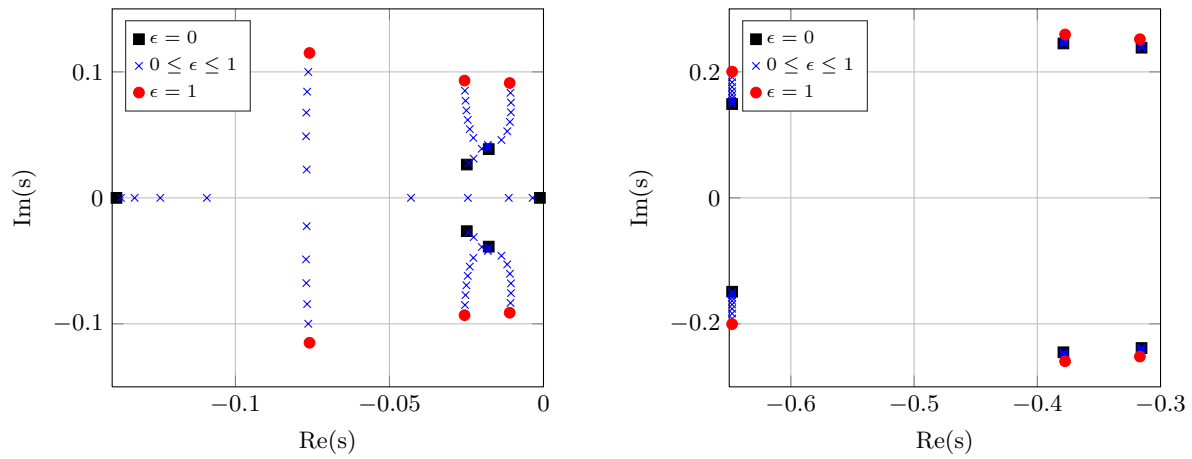
However, what happens if both spacecraft have the same size? Does the increased interaction between the two bodies affect the stability of the system? For this analysis both the target and chaser states have to be included in the state-space system. This system is then linearised and the locations of the poles are evaluated. Using a spring constant of 10 Nm/rad, the poles of the system can be seen in Fig. 5. Because of the increased number of states of the system, the number of poles has increased as well. It is now a lot harder to distinguish which paths the poles follow; however, the end result is more important.

As before, the uncontrolled system is unstable during its evolution, because there are poles in the right half-plane when the system is fully connected (Fig. 5a). Moreover, the system is already unstable when it is unconnected. However, unlike before, the controlled system now also has some unstable poles, see Fig. 5b.



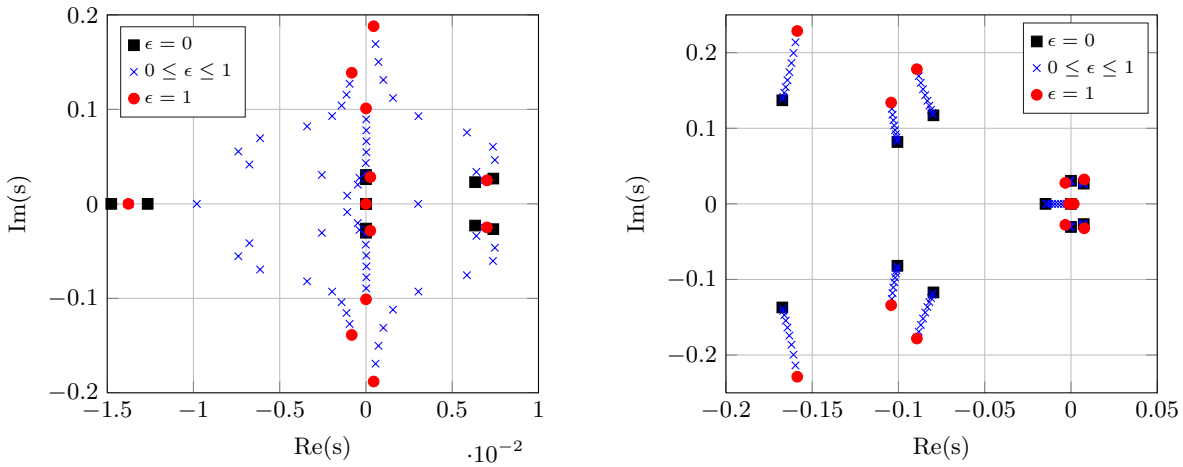
(a) $K = 10$ Nm/rad (b) $K = 1000$ Nm/rad

Figure 3: Pole location of an uncontrolled linear evolving system with large target debris



(a) Low control gains (b) High control gains

Figure 4: Pole location of a linear, evolving system with LQR control, $K = 10$N/rad, and large target debris



(a) Uncontrolled

(b) Controlled

Figure 5: Pole location of a linear, evolving system with equally sized target debris, $K = 10$ Nm/rad

What this means is that the stable controller, which has been designed for the chaser, can cause instability of the entire system when the chaser and target become connected. So, although these results are not relevant for the Envisat-chaser system, for future docking missions with smaller space debris they are relevant.

IV. Attitude Control Design

The preliminary design of the chaser consists of a Reaction Control System (RCS) and reaction wheels as actuators. Initial simulations have shown that the reaction wheels are not large enough to control the chaser during the synchronisation phase. Therefore, only the RCS will be investigated further.

The other elements of the attitude control design consist of a control algorithm to find the control moments; a control allocation method to distribute this moment among the thrusters; and a Pulse-Width Pulse-Frequency (PWPF) modulator to transform the continuous control moment into discrete pulses for the thrusters.

A. Linear Quadratic Regulator

Quaternion feedback control has seen numerous applications in spacecraft attitude systems,[5] missile control,[6] and robot manipulators.[7] The basic quaternion feedback controller has the form:

$$\mathbf{u} = -\mathbf{K}_e \mathbf{q}_e - \mathbf{C}_e \boldsymbol{\omega}_e \tag{25}$$

where $\mathbf{K}_e$ and $\mathbf{C}_e$ are gain matrices, $\mathbf{q}_e$ is the vector part (q_1, q_2, q_3) of the error quaternion q_e, and $\boldsymbol{\omega}_e$ the angular velocity error. Only the vector part of the quaternion needs to be considered, because the scalar part of the quaternion, q_4, is dependent on the vector part (through the unit-norm constraint) and does not provide any additional information to the control algorithm. The quaternion and angular-velocity error are given by:

$$q_e = q_C \otimes q_T^*, \qquad \boldsymbol{\omega}_e = \boldsymbol{\omega}_C - \boldsymbol{\omega}_T \tag{26}$$

with q_C the chaser quaternion and q_T^* the conjugate of the target quaternion.

A linear control law is considered, and Eq. (25) is rewritten as:

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \tag{27}$$

The infinite-time LQR is an optimal controller of the form of Eq. (27) for an LTI system that minimises the

quadratic performance index:

$$J = \int_0^\infty \left(\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}\right) dt \tag{28}$$

where $\mathbf{Q}$ is an $n \times n$ symmetric positive semi-definite matrix and $\mathbf{R}$ is an $m \times m$ symmetric positive definite matrix. A typical choice for $\mathbf{Q}$ and $\mathbf{R}$ is based on Bryson's rule:

$$\begin{gathered} \mathbf{Q} = \operatorname{diag}\left(q_1, q_2, ..., q_n\right), \qquad \mathbf{R} = \rho \cdot \operatorname{diag}\left(r_1, r_2, ..., r_m\right), \\ q_i = \frac{1}{\Delta x_i{}^2}, \qquad r_i = \frac{1}{\Delta u_i{}^2} \end{gathered} \tag{29}$$

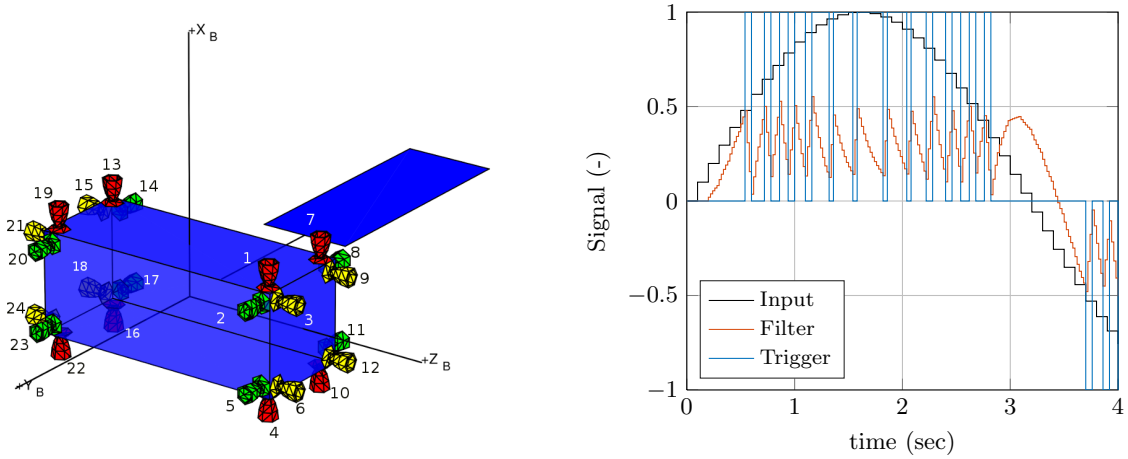
with Δx_i the permissible error in the states, Δu_i the maximum control input, and ρ a positive constant.

Initial simulations showed that a single set of gains for all three phases of the docking (unconnected, semi-connected, and stack) was not sufficient to stabilise the system. Therefore, two sets of gains were used; one for the unconnected and semi-connected phase, and one for the stack phase. These gains were computed using two different linearised systems (one with the chaser inertia tensor, and the other with the stack inertia tensor). For both sets of gains the same settings were used to compute the optimal gains:

$$\begin{gathered} \Delta q = \begin{bmatrix} 0.01 & 0.01 & 0.01 \end{bmatrix}^T, \qquad \Delta \boldsymbol{\omega} = \begin{bmatrix} 0.01 & 0.01 & 0.01 \end{bmatrix}^T \text{ deg/s} \\ \Delta \mathbf{u} = \begin{bmatrix} 100 & 100 & 100 \end{bmatrix}^T \text{ Nm}, \qquad \rho = 50 \end{gathered} \tag{30}$$

B. Control Allocation

The chaser has an RCS, which consists of 24 thrusters, providing both the translational and rotational control of the chaser. The thrusters are placed in pods of three on the vertices of the chaser, see Fig. 6a. Clearly, there is a large number of actuators and the control effort has to be distributed among the multiple, redundant actuators.



(a) Location of thrusters

(b) Discrete PWPF modulator outputs

Figure 6: Thruster subsystem

In general, the transformation between the desired control actions $\mathbf{u} = [u_1 \dots u_m]^T$ and the delivered control input to the actuators $\mathbf{a} = [a_1 \dots a_n]^T$ can be written as:

$$\mathbf{u} = \mathbf{T}\mathbf{a} \tag{31}$$

where $\mathbf{T}$ is the $m \times n$ configuration matrix. In case of the thruster set-up of the chaser this would be a 6×24 matrix. And $\mathbf{a}$ is usually limited:

$$\mathbf{a}_{\min} \leq \mathbf{a} \leq \mathbf{a}_{\max} \tag{32}$$

If $m < n$, the problem of Eq. (31) is an underdetermined system, which has either zero or infinitely many solutions. One of the methods to solve this problem is by using the Moore-Penrose pseudo-inverse. If the system has multiple solutions, the pseudo-inverse can be used to find a solution that minimises the l_2 norm of $\mathbf{a}$.[8]

The pseudo-inverse is just one of the possible methods to solve the problem of Eq. (31). In general, the fundamental control allocation problem can be formulated as finding a vector $\mathbf{a}$ such that the performance index

$$J = ||\mathbf{Ta} - \mathbf{u}|| \tag{33}$$

is minimised, subject to the constraints of Eq. (32). If $J = 0$ the problem is solved exactly. However, the solution is not necessarily unique (unless $\mathbf{T}$ satisfies certain specific conditions). An additional objective could then be, e.g., to minimise the control effort.

However, instead of solving the two problems sequentially, a mixed optimisation problem can be defined, which combines the error and control minimisation problems into a single problem. The performance index is then defined as[9]

$$J = ||\mathbf{Ta} - \mathbf{u}|| + \lambda||\mathbf{a} - \mathbf{a}_p|| \tag{34}$$

where $\mathbf{a}_p$ is some preferred control input (e.g., a nominal reaction-wheel rotational rate) and λ a weight factor. Because error minimisation has a higher priority than control minimisation, this number is chosen to be small.

Last, the order of the norm in Eq. (34) has to be specified. Bodson and Frost consider an l_1, l_2 and mixed $l_1 - l_\infty$.[9] The norm of a vector $\mathbf{x}$ can be written as:

$$||\mathbf{x}||_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \tag{35}$$

For $p = 1$ we get the l_1 norm, $p = 2$ gives the Euclidean norm, and we get the maximum norm l_∞ when p approaches infinity.

We choose the l_1-norm, because it will minimise the sum of the total control effort; a measure of the required propellant. So, the control allocation problem has been translated into a *mixed* l_1 minimisation problem. Solving such a linear programming problem can be done with standard numerical methods. In our research a self-dual parametric simplex method has been used.[10]

C. Pulse-Width Pulse-Frequency Modulation

The reaction-jet thrusters have a binary state, i.e., on or off. To deliver the required continuous control moments, some form of modulation has to be applied. In Ref. 11, a Pulse-Width Pulse-Frequency modulator is described, which consists of a first-order lag filter and a Schmitt trigger inside a feedback loop. When the filter reaches a certain threshold U_{on} the trigger outputs the maximum value U_m of the thrusters. The trigger is "reset" when the filter drops below U_{off}. The working principle of the PWPF has been visualised for a sine wave input in Fig. 6b.

A discrete time implementation of the PWPF modulator has the following transfer function:

$$H(z) = \frac{K_m \Delta T_m}{1 + (\Delta T_m - 1)z^{-1}}, \qquad \Delta T_m = \frac{\Delta t}{T_m} \tag{36}$$

where Δt is the sample time of the thrusters and T_m and K_m are filter coefficients.

V. Simulation Results

A number of simulations has been performed to analyse the stability of the system. First, for a nominal case, the transient and steady-state behaviour of the system during the unconnected, semi-connected, and

stack phase have been assessed. Second, the influence of uncertainties in the angular rate of the target have been investigated by means of a Monte-Carlo analysis.

As described in Sec. II.A, the docking phase will start with a synchronisation phase. As a nominal case, it is assumed that the target has an initial rotational rate of 3.5 deg/s with equal components around the three axes, i.e.:

$$\boldsymbol{\omega}_{T,0} = 3.5\frac{\mathbf{x}}{|\mathbf{x}|} \text{ deg/s}, \qquad \mathbf{x} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T \tag{37}$$

The target is given a zero initial attitude $q_{T,0} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$ and the chaser is assumed to be in its nominal attitude which has a roll, pitch, and yaw of 0, -90, and -90 deg, respectively, or $q_{C,0} = \begin{bmatrix} -0.5 & -0.5 & -0.5 & 0.5 \end{bmatrix}^T$, and zero angular velocity. So the chaser has to match both its attitude and angular velocity to that of the target to synchronise the rotational motion. For now, ideal navigation is assumed, yielding perfect knowledge about the state.

The error quaternion between the chaser and target is given by $q^{C/T}$. To clearly present the results in this paper, the rotation angle θ associated with this error quaternion is used to indicate the angular error the chaser and target, i.e.:

$$\theta = 2\cos^{-1}\left(q_4^{C/T}\right) \tag{38}$$

A. Synchronisation

Figure 7a shows the angle θ over time. It can be deduced that the LQR is able to reduce the angular error between the chaser and target to zero in little over 20 seconds and that the steady-state error is kept below 0.8 deg. The mean steady-state error over the last 50 seconds is equal to 0.49 deg. The desired control moments, which come directly from the LQR, are shown in Fig. 7b. This figure reveals that the control effort is high at the beginning of the synchronisation phase. Furthermore, the thrusters have to fire continually to keep the chaser aligned with the target, which has a constantly changing angular velocity vector.

The actual control moments generated by the thrusters are shown in Fig. 7c, where each spike corresponds to one or more thruster firings. Comparing Fig. 7b and 7c, the effects of the control-allocation algorithm and the PWPF modulator are clearly visible. Using an I_{sp} of 230 sec, the total propellant mass is calculated to be 0.64 kg.

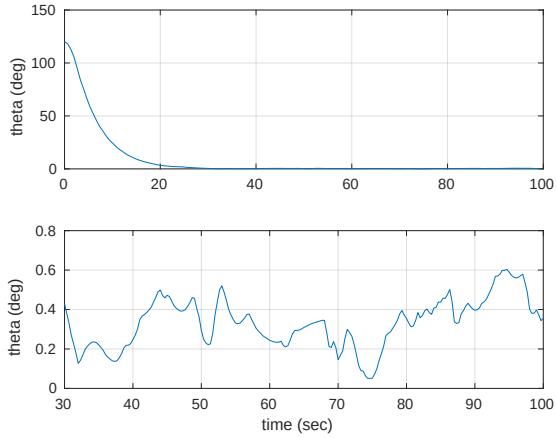
The angular velocity of the target is assumed to be uncertain. Therefore, a Monte Carlo analysis has been performed on the synchronisation phase to see what the effect of varying angular velocities is on the steady-state error and propellant usage of the chaser. To this end, the angular velocity of Eq. (37) is allowed to take on any value, as long as it has a magnitude of 3.5 deg/s.

In Fig. 8 the results of a Monte Carlo run with 1000 samples is shown. The mean steady-state error (Fig. 8a) is the mean value of the error over the last 50 seconds of the simulation. As can be seen, the steady-state error remains below 0.8 deg for all initial angular velocities, and its mean is 0.38 deg. Furthermore, the total propellant (Fig. 8b) has a mean value of 0.72 kg.

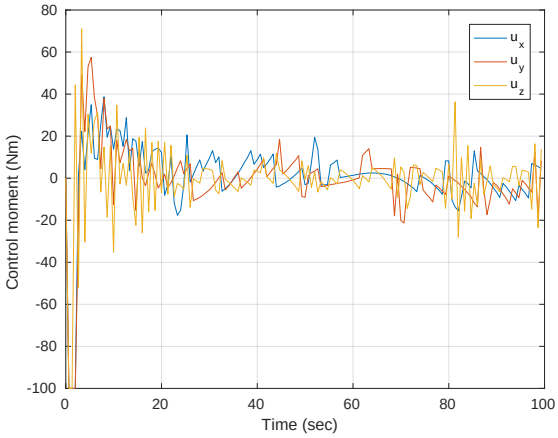
B. Semi-connected

An important aspect for the simulation of the semi-connected phase is the time-history of the evolutionary parameter, ϵ. Therefore, a simple assessment has been made about the time line of the docking. Initially, there is no connection, meaning that $\epsilon = 0$, then, as the robot arm grasps the target, the stiffness will slowly start increasing. Then, as the tentacles and pushing rods are deployed, the stiffness is expected to quickly rise. As can be seen in Fig. 9 this has been modelled by taking two slopes. When the robot arm grasps the target, the system is still flexible (a gradual slope for 50 seconds), but as soon as the tentacles close and the pushing rods are deployed, the system becomes less flexible (a steeper slope for 50 seconds). For analysis purposes the system is then assumed to remain at the same flexibility for another 50 seconds, while in reality the system would become one (rigid) body. This has been done to see what the long-term effect of the connection is on the (numerical) stability of the system. The spring constant has been assumed to be equal to 1,000 Nm/rad around all three axes.

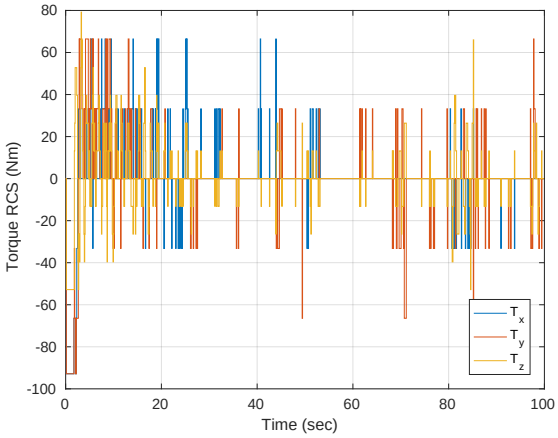
At the beginning of the semi-connected phase, the state of the chaser and target are the same as at the end of the synchronisation phase. Then, as can be seen from Fig. 10a, the error between the two spacecraft decreases when the connection becomes stiffer. As expected from the linear stability analysis, the system



(a) Error and zoom-in

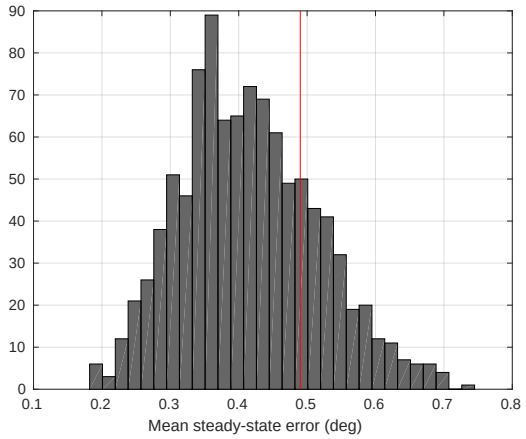


(b) Desired control moments

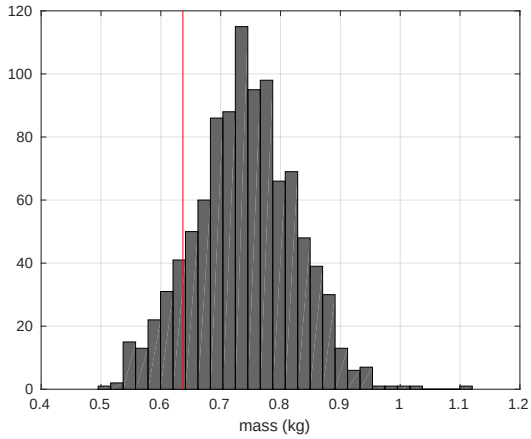


(c) Actual control moments

Figure 7: Synchronisation with LQR



(a) Steady-state error



(b) Total propellant use

Figure 8: Monte Carlo runs (1000) for stack with varying angular velocity

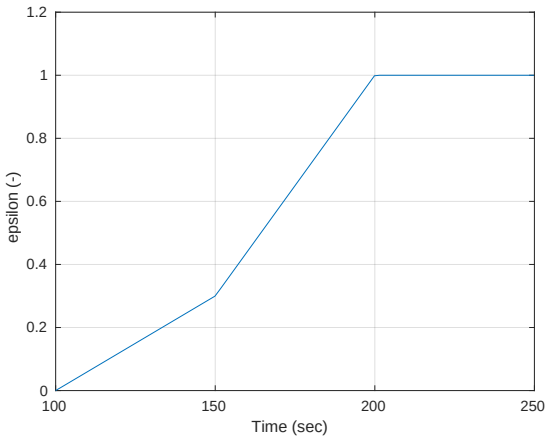
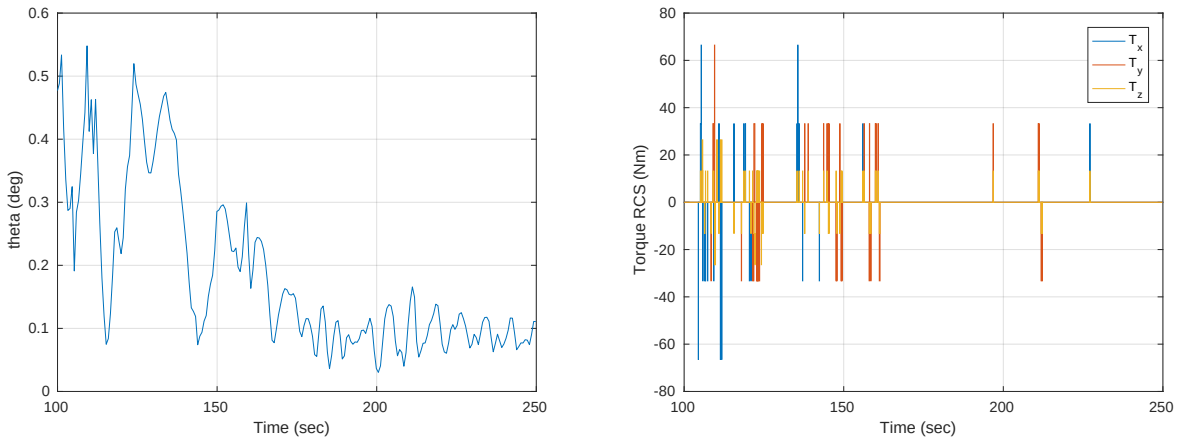


Figure 9: Evolution parameter

remains stable. Furthermore, the steady-state error decreases to 0.1 deg. Thus, the connection has little impact on the motion of the target and it actually decreases the error between the chaser and target. As a result, less control effort is required, as can be seen in Fig. 10b. The total propellant required during this phase is 0.12 kg.



(a) Error

(b) Control effort

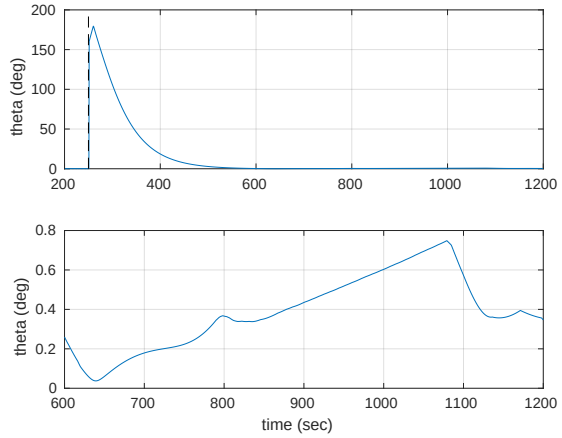
Figure 10: Semi-connected phase with LQR

C. Stack phase

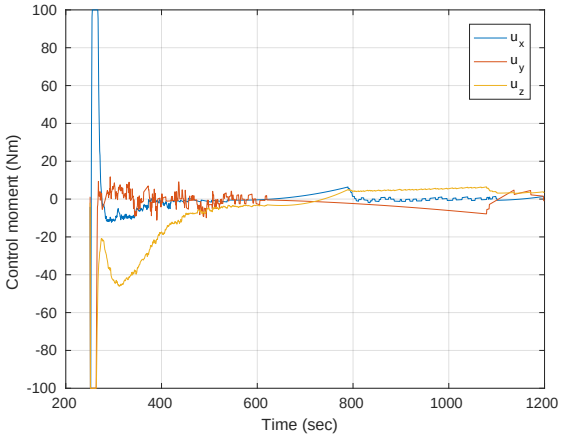
Once the two spacecraft have docked, it is assumed that they have formed one, single, rigid body. The objective of the control algorithm then switches to stabilising and reorienting the stack. To this end, a new commanded attitude is given, which has zero angular velocity and a roll, pitch, and yaw angle of -90, 0, and 0 deg, respectively. In this attitude the stack is gravity-gradient stabilised, its deorbit thrusters are pointed in the correct direction to slow down the stack, and a direct line of communication with the Earth is possible.[12]

As can be seen in Fig. 11a, the LQR is again able to reduce the (new) error to zero. Because of the large size of the stack, the required time is now roughly 250 seconds. Furthermore, the steady-state error remains below 0.8 deg with the mean steady-state error over the last 400 seconds equal to 0.52 deg. The control effort during the transient response is a lot larger than it was during the two previous phases (Fig. 11b). The

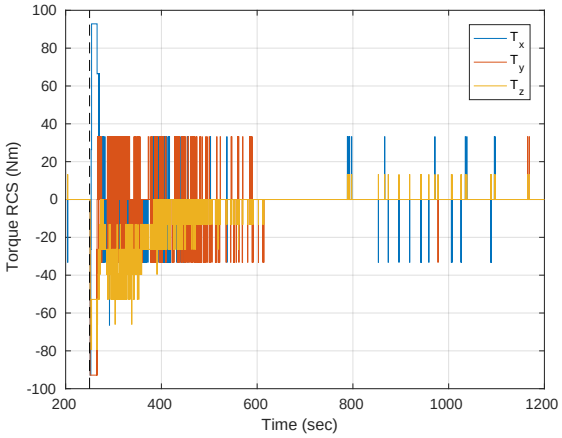
actuators remain saturated for a longer period and the control effort is large till $t = 600$. After that, less control is required to keep the stack stable, because the objective of the controller is now to keep the stack in a certain static attitude. It only has to compensate for the primary disturbance: the gravity-gradient torque. The total required propellant during the stack phase is equal to 4.0 kg.



(a) Error and zoom-in



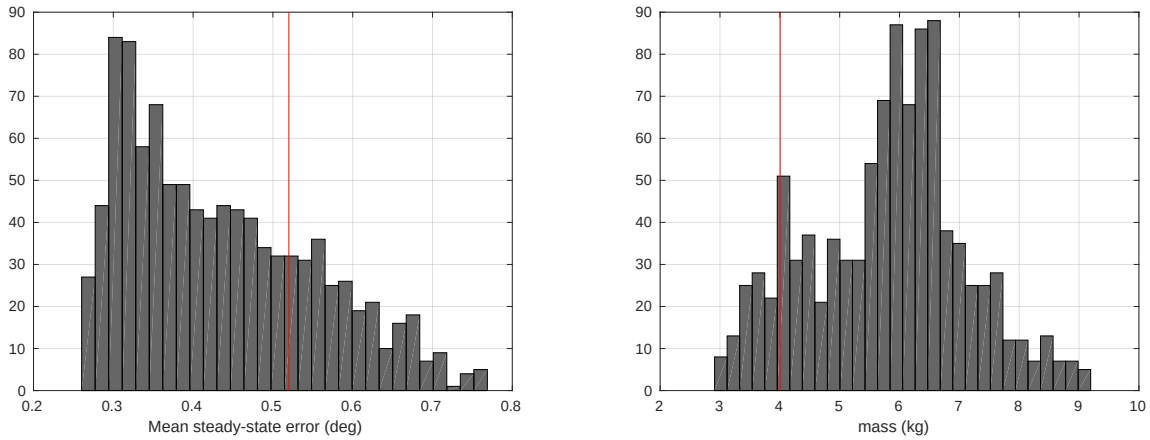
(b) Desired control moments



(c) Actual control moments

Figure 11: Stack reorientation

Again, a Monte-Carlo analysis was performed to see if the LQR is capable of stabilising the target for varying initial angular velocities. As can be seen in Fig. 12a, the steady-state error remains below 0.8 deg as well and has a mean of 0.4 deg. The propellant usage has increased to a mean value of 5.8 kg, see Fig. 12b. This value is much larger than for the synchronisation phase, because the stack system has a larger inertia and the simulation period was longer.



(a) Steady-state error

(b) Total propellant use

Figure 12: Monte Carlo runs (1000) for stack with varying angular velocity

VI. Conclusions and Recommendations

In this paper the rotational motion of a chaser spacecraft that docks with large space-debris, was investigated. The Evolving-Systems framework was applied to the docking problem; using a simple spring-damper the connection between the two spacecraft was modelled and the progression of the docking was captured in an evolutionary parameter.

First, the linear stability of the system was analysed by looking at the poles of the linearised system. For a system with large space-debris, i.e., the inertia of the debris is two orders of magnitude larger than that of the chaser, the controlled system remains stable. However, for a system with equally-sized objects, instability can occur even when the chaser is actively controlled. For future debris removal mission this could be of relevance when the debris is relatively small.

Next, the nonlinear behaviour of the system was investigated. Using a simple Linear Quadratic Regulator, it was found that the system remains stable before, during, and after docking with the large debris. The chaser synchronises with the target in 25 seconds, the steady-state error is 0.5 deg, and the propellant usage is 0.64 kg. During the docking, the error decreases to 0.1 deg and requires only 0.12 kg of propellant. Again, it was found that the effect of the docking has little impact on the motion of the system, because of the large size of the debris. The stack is able to reorient in 250 seconds and reaches a steady-state error of 0.52 deg, requiring 4.0 kg of propellant.

Furthermore, even for varying initial angular velocities the LQR could stabilise the system. Per ESA's requirements, the angular velocity vector can have any direction as long as it has a magnitude of 3.5 deg/s. The synchronisation phase has a mean steady-state error of 0.38 deg and requires an average of 0.72 kg of propellant. The stack mean steady-state error is equal to 0.4 deg and the mean propellant usage is 5.8 kg.

Evolving Systems proved to be a useful tool to evaluate the linear stability of the system. However, for the nonlinear system a more detailed analysis should be performed. First, the simple spring-damper is too much of an approximation to model the complex dynamics of the robot arm, tentacles, and pushing rods. This improved model might involve multiple, interconnected spring-dampers. Second, the time-history of the evolutionary parameter has to be studied in more detail. For now, a linear relation was used, but there is no evidence that the actual docking procedure proceeds in this fashion.

References

[1]Klinkrad, H., *Space Debris: Models and Risk Analysis.*, Springer, Berlin, 2006.

[2]ESA, "CDF Study Report e.Deorbit", Noordwijk, 2012, [CDF-135(C)].

[3]Frost, S.A. and Balas, M. "Evolving Systems and Adaptive Key Component Control", in: *Aerospace Technologies Advancements*, InTech, Rijeka, 2010, pp. 115-140.

[4]Frost, S.A. and Balas, M.J., "Evolving systems: Adaptive key component control and inheritance of passivity and dissipativity", *Applied Mathematics and Computation*, Vol. 217, No. 3, 2010, pp 1034-1044.

[5]Wie, B. and Barba, P.M., "Quaternion Feedback for Spacecraft Large Angle Maneuvers", *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 3, 1985, pp. 360-365

[6]Song, C. and Kim, S. and Kim, S. and Nam, H.S., "Robust Control of the Missile Attitude based on Quaternion Feedback", *Control Engineering Practice*, Vol. 14, No. 7, 2006, pp. 811-818

[7]Xian, B. and de Queiroz, M.S. and Dawson, D. and Walker, I., "Task-Space Tracking Control of Robot Manipulators via Quaternion Feedback", *Robotics and Automation*, Vol. 20, No. 1, 2004, pp. 160-167

[8]Ben-Israel, A. and Greville, T.N.E., "Generalized Inverses: Theory and Applications", Springer-Verlag, New York, 2003, Ed. 2

[9]Bodson, M. and Frost, S.A., "Load Balancing in Control Allocation", *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 2, 2011, pp. 380-387.

[10]Vanderbei, R.J., *Linear Programming: Foundations and Extensions*, 2001, Springer, New York, Ed. 2.

[11]Wie, B. and Plescia, C.T., "Attitude Stabilization of Flexbile Spacecraft During Stationkeeping Maneuvres", *Journal of Guidance, Control, and Dynamics*, Vol. 7, No. 4, 1984, pp. 430-436

[12]Habets, J.M.G., "Evolving Systems Approach to the Attitude Control of a Large Space Debris Removal Spacecraft", M.Sc. Thesis, TU Delft, 2015, Available at: http://repository.tudelft.nl.