Constrained Low-Thrust Satellite Formation-Flying Using Relative Orbit Elements

Autonomous Guidance and Control of the NetSat Satellite Formation-Flying Mission

Lukas Maximilian Steindorf

Space Engineering, masters level (120 credits)
2017

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
Constrained Low-Thrust Satellite Formation-Flying Using Relative Orbit Elements

Autonomous Guidance and Control for the NetSat Satellite Formation-Flying Mission

Lukas M. Steindorf

Master's Thesis
A thesis submitted for the degree Teknologien Masterexamen med Huvudområde Rymdteknik (MSc with a Major in Space Technology) at Luleå University of Technology and Master of Science (MSc) at the University of Würzburg
Constrained Low-Thrust Satellite Formation-Flying Using Relative Orbit Elements

Autonomous Guidance and Control for the NetSat Satellite Formation-Flying Mission

by

Lukas M. Steindorf

to obtain the degree of Master of Science
at the University of Würzburg,
at the Luleå University of Technology,
to be defended publicly in B202 in the Department of Computer Sciences of the University of Würzburg on Thursday December 15, 2016 at 05:00 PM.

Student numbers:  2054126 (University of Würzburg)
                  910519  (Luleå University of Technology)
                  6108026 (Stanford University)

Project duration: June 1, 2016 – December 1, 2016

Thesis committee: Prof. Dr. S. D’Amico, Stanford University
                  J. Scharnagl, Zentrum für Telematik
                  Prof. Dr. K. Schilling, University of Würzburg, examiner
                  Prof. Dr. R. Emami, Luleå University of Technology, examiner

An electronic version of this thesis is available at http://repository.ltu.se/.
Affidavit

I hereby confirm that my thesis entitled “Constrained Low-Thrust Satellite Formation-Flying Using Relative Orbit Elements” is the result of my own work. I did not receive any help or support from commercial consultants. All sources are applied and specified in the thesis. Furthermore, I confirm that this thesis has not yet been submitted as part of another examination process neither in identical nor in similar form.

Lukas M. Steindorf
Stanford, November 2016
Abstract

This thesis proposes a continuous low-thrust guidance and control strategy for satellite formation-flying. Stabilizing feedback based on mean relative orbit elements and Lyapunov theory is used. A novel feedback gain matrix inspired by the fuel-optimal impulsive solution is designed to achieve near-optimal fuel consumption. A reference governor is developed to autonomously guide the spacecraft through the relative state-space in order to allow for arbitrarily constrained satellite formations. Constraints include desired thrust levels, time constraints, passive collision avoidance and locally constrained state-space areas. Keplerian dynamics are leveraged to further decrease fuel consumption. Simulations show fuel consumptions of only 4% higher delta-$v$ than the fuel-optimal impulsive solution. The proposed control and guidance strategy is tested in a high-fidelity orbit propagation simulation using MATLAB/Simulink. Numerical simulations include orbit perturbations such as atmospheric drag, high-order geopotential, solar radiation pressure and third-body (Moon and Sun) effects. Test cases include reconfiguration scenarios with imposed wall, thrust and time constraints and a formation maintenance experiment as flown by TanDEM-X, the TanDEM-X Autonomous Formation-Flying (TAFF) experiment.
## Contents

1 Introduction ................................................. 1  
1.1 Approach ............................................. 2  

2 Quasi-Nonsingular Relative Orbit Elements ............... 5  

3 Linear Dynamic Model .................................. 7  
3.1 Reduced Model ...................................... 7  

4 Control Strategy .................................... 9  
4.1 Stabilizing feedback .................................. 9  
  4.1.1 Continuous Low-Thrust Control in Near-Circular Orbits ......... 9  
  4.1.2 Continuous Low-Thrust Control in Eccentric Orbits ........... 10  
4.2 Controlling the Along-Track Separation Rate ............ 11  

5 Constraints ............................................. 15  
5.1 Time constraint ..................................... 15  
5.2 Thrust Constraint ................................... 18  
5.3 Circular No-Entry-Zone Constraint .................... 18  
5.4 Passive Collision Avoidance ......................... 18  
5.5 Wall Constraint ..................................... 18  

6 Guidance Strategy - Reference Governor ............... 19  
6.1 The Global Potential Field and its Gradient ............ 21  
6.2 Lyapunov Threshold for the Time Constraint .......... 23  
6.3 Lyapunov Threshold for the Thrust Constraint ......... 23  
6.4 Wall Constraint ..................................... 24  
  6.4.1 Lyapunov Threshold ............................... 24  
  6.4.2 Gradient of the Potential Field .................. 24  
6.5 Circular No-Entry-Zone Constraint .................... 25  
  6.5.1 Lyapunov Threshold ............................... 25  
  6.5.2 Gradient of the Potential Field .................. 26  
6.6 Passive Collision Avoidance Constraint ............... 26  
  6.6.1 Lyapunov Threshold ............................... 26  
  6.6.2 Potential Field ................................... 27  

7 Implementation ........................................ 29  
7.1 Initialization ........................................ 29  
7.2 Propagator .......................................... 31  
  7.2.1 Pseudo Measurement and State Estimation ............. 31  
7.3 Feedback Control .................................... 34  
  7.3.1 Reference Governor ............................... 37  
7.4 Thruster Block ........................................ 47
8 Examples  49
  8.1 Thrust Constraint and Optimality. 49
  8.2 Reconfiguration with Wall Constraints  53
  8.3 Formation Maintenance (TAFF)  55
  8.4 Time Constraint  56
9 Conclusion and Way Forward  59
Bibliography  61
A Matrices  63
B Derivation of the Out-of-Plane Delta-v Lower Bound  67
1
Introduction

Space Science and Technology is currently undergoing a change to distributed mission architectures to reduce development costs, development time and enabling possibilities for high fidelity experiments and new mission concepts. Often the relative motion of cooperating spacecrafts has to be reconfigured while collisions have to be precluded. While all flown spacecraft formation-flying missions so far have only consisted of two spacecrafts, except for the NASA Magnetospheric Multiscale Mission [20] that flies four spacecrafts in a tetrahedral formation with the smallest distance of 10km at apogee. The German research institute Zentrum für Telematik and the University of Würzburg are successfully engaging into satellite design and operation since 2005 [18]. The University Würzburg’s experimental (UWE) satellites were kicked off with the UWE-1 satellite, the first German cubesat in operation [2]. UWE-1 established a telecommunication link before its follow-up mission UWE-2 proved its attitude and orbit determination capabilities in 2009 [19]. The first attitude control was demonstrated with UWE-3 in 2013 [4, 18]. UWE-3 performs autonomous attitude determination and control in real-time. The current cubesat in development is the UWE-4, which will be equipped with a low-thrust electrical propulsion system for orbit control. The NetSat mission is a technology demonstration aimed to fly satellite formations consisting of four cubesats, which allow 3D-formations (e.g. cartwheel formations for digital elevation models). The first milestone will be the NetSat-0 mission, which will launch an upgraded UWE-4 version in close proximity to the - by that time - already orbiting UWE-4 spacecraft. NetSat-0 shall demonstrate relative navigation capabilities before the next missions will show formation guidance, navigation and control.

This thesis addresses the problem of guidance and control for satellite formation-flying missions using low-thrust actuators. The fundamental goal is to establish autonomous reconfiguration of arbitrarily constrained satellite formations with near-optimal fuel consumption. Constraints include passive collision avoidance, desired thrust levels, performing a reconfiguration under time constraints and locally constrained state-space areas. With the miniaturization trend of spacecraft and electric propulsion systems, thrust levels are now found in the µN range. This requires control to be applied continuously, a strategy prone to high fuel use. Therefore this thesis proposes a first-of-its-kind continuous low-thrust guidance and control strategy with a new approach to minimize fuel consumption. The strategy is to be implemented in the NetSat mission of the University of Würzburg.
NetSat intends to prove the feasibility to operate a formation of four cubesats as a clear enabling technology for scientific experiments, as well as commercial applications.

In contrast to the available body of literature, which primarily focuses on near-circular applications of satellite formations, this paper presents a control strategy that is applicable to closed orbits of arbitrary eccentricity. Autonomous low-thrust reconfiguration guidance of constrained satellite formations is addressed in relative orbit element state-space. More specifically, the contributions to the state-of-the-art of this thesis are 1) a stabilizing feedback control law based on mean relative orbit elements (ROE) and Lyapunov theory, 2) a feedback gain matrix, which enables the spacecraft to autonomously apply thrust at fuel efficient locations, 3) the design of a reference governor that guides the spacecraft through the ROE state-space in order to ensure that all constraints imposed on the formation-flying mission are satisfied at all times and 4) an autonomous algorithm to leverage Keplerian dynamics to achieve near-optimal fuel consumption.

Continuous satellite formation control literature explores multiple approaches, including proportional derivative [16], model predictive [9], linear quadratic [14], Lyapunov [17] and adaptive [15] controllers. These control strategies largely assume near-circular reference orbits and allow arbitrary thrust vectors (not fuel efficient). Furthermore, optimality is at its best guaranteed with respect to sub-optimal cost functions (linear quadratic control). This work is primarily inspired by the fuel-optimal impulsive solution using the mean ROE description and an explicit reference governor developed for unmanned aerial vehicles. In particular, the research of Chernick et al. [5] and Nicotra et al. [13]. The research presented in this thesis provides a feasible control strategy based on Lyapunov theory in mean ROE state-space and an application of a reference governor based on potential fields.

1.1. Approach

First, a linear dynamic model valid for closed orbits of arbitrary eccentricity is developed in mean quasi-nonsingular relative orbit element state-space that makes use of the work performed by D’Amico et al. [5, 11, 21]. Inspired by results of the fuel-optimal impulsive solution [5] the control strategy does not make use of radial thrust in the co-rotating orbital frame centered at the reference spacecraft. Thus, a reduced linear dynamic model is developed that only allows thrust into the along-track and normal orbit directions. The along-track separation is controlled by leveraging Keplerian orbit dynamics, a difference in the semi-major axis leads to a relative drift of the along-track separation. This is leveraged to achieve an absolute change of the along-track separation by controlling the difference in the semi-major axis. Furthermore, the state is augmented with the rate of change of the difference in the semi-major axis in order to account for differential drag. A feedback controller based on Lyapunov theory is developed with a novel feedback gain matrix, which autonomously feeds back tracking errors at fuel-efficient locations separated by half a orbit period to stabilize the system at the applied reference. Furthermore, the feedback gain matrix is defined in a way that allows the thrust inputs to be centered around the fuel-optimal locations.
Next, a reference governor is developed in relative orbit element state-space. The main tasks of the reference governor are 1) to guide the relative spacecraft state through the ROE state-space to a desired reference and 2) to enforce and satisfy all constraints imposed on the relative spacecraft state. The first task is accomplished using attractive (for the desired reference) and repulsive (for the constraint boundaries) potential fields. Thus, a potential field map is generated which is used by the reference governor to guide the applied reference to the desired reference. The second task is realized by translating all constraints imposed on the relative state into a single constraint, called the Lyapunov threshold. The reference governor ensures that all constraints are satisfied as long as the Lyapunov threshold is not violated. Potential fields and Lyapunov thresholds are developed for arbitrary wall constraints (locally prohibited state-space areas), the thrust constraint and for the passive collision avoidance constraint. The thrust constraint is important such that thrusters are not operated in saturation and to make the controller predictable. Thus, the desired thrust peak value can be defined as a function of the desired reconfiguration time.

Finally, the guidance and control strategy is numerically validated using a high-fidelity simulation subject to orbit perturbations such as atmospheric drag, high-order geopotential, solar radiation pressure and third-body (Moon and Sun) effects. The true relative state positions and velocities are corrupted by realistic GPS sensor noise, that is then filtered by an Extended Kalman Filter. An actuator model represents a low-thrust Field Emission Electric Propulsion (FEEP) system [3] that is currently being developed by the Universities of Würzburg and Dresden and is going to be used on the NetSat mission. Results show that delta-$v$ consumption is about 4% higher than the impulsive lower-bound and the reconfiguration time is well predictable and therefore adjustable. Overall, the work presented in this thesis enables reasonable continuous low-thrust guidance and control of constrained and perturbed satellite formations in arbitrary closed orbits. It further paves the way for autonomous, active collision avoidance and reconfiguration guidance of satellite swarms.
Quasi-Nonsingular Relative Orbit Elements

This thesis deals with the relative motion of two spacecraft. One spacecraft is defined to be the chief spacecraft and the other one is defined to be the deputy spacecraft. The only difference between both spacecraft in the context of this thesis is the fact the chief spacecraft is the center spacecraft. The motion of the deputy spacecraft is then described with respect to the chief/center spacecraft and is expressed within the RTN frame. The RTN orbital frame (see figure 2.1), which is located at the position of the center spacecraft and is composed of unit vectors in radial direction $\mathbf{R} = \hat{r} = \frac{r}{|r|}$, normal direction $\mathbf{N} = \hat{n} = \mathbf{R} \times \frac{v}{|v|}$ and the along track direction $\mathbf{T} = \hat{t} = \mathbf{N} \times \mathbf{R}$ that is tangential to the orbit for $e = 0$. This frame is also referred to as to co-rotating orbital frame. The absolute orbits of the chief and the deputy spacecraft are defined by mean classical orbit elements $\alpha$ (semi-major axis $a$, eccentricity $e$, inclination $i$, argument of the ascending node $\Omega$, argument of the periapsis $\omega$ and the mean anomaly $M$)

$$\alpha = \begin{pmatrix} a \\ e \\ i \\ \Omega \\ \omega \\ M \end{pmatrix}.$$  

(2.1)

The relative motion can conveniently be described using nonlinear combinations of the mean classic orbit elements $\alpha$ of the chief and deputy spacecraft. This combination is the quasi-nonsingular relative orbit element (ROE) set $\delta \mathbf{a}_{qns}$ and is given by [6]

$$\delta \mathbf{a}_{qns} = \begin{pmatrix} \delta a \\ \delta \lambda \\ \delta e_x \\ \delta e_y \\ \delta i_x \\ \delta i_y \end{pmatrix} = \begin{pmatrix} (a - a_c)/a_c \\ u - u_c + (\Omega - \Omega_c) \cos i_c \\ e_x - e_{xc} \\ e_y - e_{yc} \\ i - i_c \\ (\Omega - \Omega_c) \sin i_c \end{pmatrix},$$  

(2.2)

where the subscript $(.)_c$ denotes the chief spacecraft. Throughout this thesis $\delta(.)$ denotes a relative orbit element and $\Delta(.)$ an actual arithmetic difference or error. The parameters $e_x$ and $e_y$ are defined by $e_x = e \cos \omega$ and $e_y = e \sin \omega$, the mean argument of latitude is given by $u = \omega + M$. The first component of the ROE state is normalized by the semi-major
axis of the chief spacecraft. Thus, a non-dimensional state is created, which is composed of the relative semi-major axis \( \delta a \), the relative mean longitude \( \delta \lambda \), the eccentricity vector \( \delta e = e - e_c \) and the inclination vector \( \delta i = i - i_c \). The vectors \( \delta e \) and \( \delta i \) are defined by

\[
\delta e = \begin{pmatrix}
\delta e_x \\
\delta e_y
\end{pmatrix} = \begin{pmatrix}
\delta e \cos \varphi_{\text{tp}} \\
\delta e \sin \varphi_{\text{tp}}
\end{pmatrix}, \quad \delta i = \begin{pmatrix}
\delta i_x \\
\delta i_y
\end{pmatrix} = \begin{pmatrix}
\delta i \cos \theta_{\text{ran}} \\
\delta i \sin \theta_{\text{ran}}
\end{pmatrix},
\]

with relative perigee \( \varphi_{\text{tp}} \) and relative ascending node \( \theta_{\text{ran}} \). ROEs allow geometric interpretation of the relative motion in the RTN orbital frame. The amplitudes of the relative motion are given by \( a_c \| \delta e \| \) in radial (R) direction, \( 2a_c \| \delta e \| \) in along-track (T) direction and \( a_c \| \delta i \| \) in out-of-plane (N) direction (see figure 2.2). Offsets in along-track and radial direction are given by \( a_c \delta \lambda \) and \( a_c \delta a \) respectively [7]. For elliptic orbits this motion is superimposed by an additional mode of twice the frequency and amplitude proportional to the eccentricity of the center spacecraft \( e_c \) [5].

Figure 2.1: RTN frame shown for the general case of a closed eccentric orbit. For \( e = 0 \) both the along-track direction T and the velocity are parallel (T = v). The right handed coordinate frame is composed from the rectangular unit vectors R, T and N.

Figure 2.2: Geometric interpretation of quasi-nonsingular ROEs for near-circular orbits. The projection of the relative motion into the rt-plane (in-plane motion) is shown on the left. The projection into the rn-plane (out-of-plane motion) is shown on the right.
The full state-space model (with three control inputs and six state outputs augmented with the differential rate of change $\delta \dot{a}$ of the differential semi-major axis $\delta a$) of the relative motion of a controlled spacecraft with respect to a reference spacecraft is given by

\[
\begin{bmatrix}
\delta \dot{a} \\
\delta \dot{\dot{a}}
\end{bmatrix} = \begin{bmatrix}
\vec{A} \\
\vec{A}
\end{bmatrix} \begin{bmatrix}
\delta \dot{a} \\
\delta \dot{\dot{a}}
\end{bmatrix} + \begin{bmatrix}
\vec{B} \\
0^{1 \times 3}
\end{bmatrix} \vec{u},
\]

(3.1)

where $\vec{u} = (u_t, u_n)^T$ is the vector of control accelerations in the RTN orbital frame. The plant matrix is given by $\vec{A} = \vec{A}_{J^2} + \vec{A}_{\text{drag}} + \vec{A}_{\text{Kepler}}$ [Eqs. (A.1), (A.3) and (A.5)]. $\vec{A}_{J^2}$, $\vec{A}_{\text{drag}}$ and $\vec{A}_{\text{Kepler}}$ [references [21] and [5]] describe the behavior of the ROEs under the influence of the Earth’s oblateness effect $J^2$, differential drag and Keplerian motion, respectively. The control input matrix $\vec{B} \in \mathbb{R}^{6 \times 3}$ is given by equation (A.7) [5]. In order to accurately include the differential aerodynamic drag into the control law it is essential to properly initialize $\delta \dot{a}$. This can be done by in-situ measurements of the differential semi-major axis by tracking the difference over a few orbits without any applied maneuvers.

### 3.1. Reduced Model

The study of the control input matrix $\vec{B}$ (equation (A.7)) shows that a change of the eccentricity vector ($\delta e$) is most efficiently achieved by applying tangential thrust only. Similar, the difference in the semi major axis $\delta a$ is most efficiently controlled by tangential thrust only. This suggests to control the system given by equation (3.1) without radial thrust. Note, that $\delta \lambda$ can only be controlled by radial thrust and therefore waiving radial thrust would mean the loss of full controllability. In this situation it is convenient to use a reduced model and to control $\delta \dot{\lambda}$ by leveraging Keplerian dynamics. This can be done by augmenting the state-space with $\delta \dot{\lambda}$ [8] or by changing the applied reference (see section 4.2) that will then be followed by a stabilizing feedback controller to maintain full controllability of the complete ROE set $\delta \vec{a}$. The reduced model is given by

\[
\begin{bmatrix}
\delta \dot{a} \\
\delta \dot{\dot{a}}
\end{bmatrix} = \begin{bmatrix}
\vec{A} \\
\vec{A}
\end{bmatrix} \begin{bmatrix}
\delta \dot{a} \\
\delta \dot{\dot{a}}
\end{bmatrix} + \begin{bmatrix}
\vec{B} \\
0^{1 \times 3}
\end{bmatrix} \vec{u},
\]

(3.2)

where the reduced ROE vector is defined as $\delta \vec{a} = [\delta a, \delta e_x, \delta e_y, \delta i_x, \delta i_y]^T$ and the control input $\vec{u} = (u_t, u_n)^T$ is the control acceleration in along-track and normal direction of the
co-rotating RTN-frame. The reduced plant matrix is now \( A = A_{f2} + A_d \) (equations (A.2) and (A.4)). Note, that Keplerian dynamics have no influence on \( \delta\alpha \). The reduced control input matrix is \( B \) (see equation A.8).
4.1. Stabilizing feedback

The feedback controller has to stabilize the system at an applied reference $\delta \alpha_a$. This applied reference is not necessarily the desired reference $\delta \alpha_{\text{ref}}$. The proposed reference governor in section 6 ensures at steady-state $\delta \alpha_a = \delta \alpha_{\text{ref}}$, if the imposed constraints on the relative state allow. A control law which ensures the relative spacecraft state to asymptotically tend the applied reference is given by

$$u = -B^* \left[ A \delta \alpha + P (\delta \alpha - \delta \alpha_a) \right]. \quad (4.1)$$

Let the system described by equation (3.2) be subject to the control law given in (4.1) with $P$ being positive definite. A Lyapunov function candidate is given by

$$V(\delta \alpha, \delta \alpha) = \frac{1}{2} (\delta \alpha - \delta \alpha_a)^T (\delta \alpha - \delta \alpha_a),$$

$$= \frac{1}{2} \Delta \delta \alpha^T \Delta \delta \alpha. \quad (4.2)$$

By taking the derivative of eq. (4.2) and substituting eq. (4.1) into eq. (4.2) one can proof that eq. (4.2) indeed is a Lyapunov function:

$$\dot{V}(\delta \alpha, \delta \alpha) = \Delta \delta \alpha^T \Delta \delta \dot{\alpha},$$

$$= \Delta \delta \alpha^T (\delta \dot{\alpha} - \delta \dot{\alpha}_a),$$

$$= \Delta \delta \alpha (A(\alpha_c) \delta \alpha + B(\alpha_c) u - 0),$$

$$= \Delta \delta \alpha \left[ A(\alpha_c) \delta \alpha + B(\alpha_c) (-B^* [A \delta \alpha + P (\delta \alpha - \delta \alpha_a)]) \right],$$

$$= -\Delta \delta \alpha^T P \Delta \delta \alpha, \quad (4.3)$$

which is negative definite. Note, that $\delta \alpha_a$ is changed during reconfigurations ($\delta \dot{\alpha}_a \neq 0$). However, for slow reconfigurations (low-thrust) and for steady state ($\delta \alpha_a = \delta \alpha_{\text{ref}}$) one can consider $\delta \alpha_a$ to be constant.

4.1.1. Continuous Low-Thrust Control in Near-Circular Orbits

To improve fuel efficiency, the Lyapunov feedback gain matrix $P$ is designed such that it determines fuel efficient locations to apply control inputs.
Control Strategy

\[ P = \frac{1}{k} \begin{pmatrix} \cos(J)^N & 0 & 0 & 0 & 0 \\ 0 & \cos(J)^N & 0 & 0 & 0 \\ 0 & 0 & \cos(J)^N & 0 & 0 \\ 0 & 0 & 0 & \cos(H)^N & 0 \\ 0 & 0 & 0 & 0 & \cos(H)^N \end{pmatrix} \]  

(4.4)

where \( k \in \mathbb{R}^+ \) is an arbitrary large scaling scalar, \( J = u - \bar{u}_{ip} \), \( H = u - \bar{u}_{oop} \) and \( u = M + \omega \) is the mean argument of latitude. The exponent \( N \in \mathbb{N} \mid N \text{ mod } 2 = 0 \land N > 2 \) defines to which extent the control inputs are centered around the fuel optimal locations (argument of latitudes) to control the in-plane \( (\delta \alpha, \delta \epsilon) \) motion at \( \bar{u}_{ip} \) and the out-of-plane \( (\delta i) \) motion at \( \bar{u}_{oop} \). The fuel optimal location can be determined as shown in [5] and according to figure 4.1. The optimal locations for near-circular orbits to apply thrust are given by

\[ \bar{u}_{ip} = \tan^{-1} \left( \frac{\Delta \delta e_y}{\Delta \delta e_x} \right) \]  

(4.5)

\[ \bar{u}_{oop} = \tan^{-1} \left( \frac{\Delta \delta i_y}{\Delta \delta i_x} \right) \]  

(4.6)

4.1.2. Continuous Low-Thrust Control in Eccentric Orbits

The fuel-optimal true argument of latitude \( \bar{\theta}_{ip} \) to apply tangential thrust in closed eccentric orbits for in-plane reconfiguration is found by numerically solving the following expression [5]

\[ \tan(\bar{\theta}_{ip}) \frac{1 + e_y}{1 + \frac{2 + e_x \cos(\bar{\theta}_{ip}) + e_y \sin(\bar{\theta}_{ip})}{\sin(\bar{\theta}_{ip})}} = \frac{\Delta \delta e_y \tan(i_c) + e_y \Delta \delta i_y}{\Delta \delta e_x \tan(i_c) - e_y \Delta \delta i_y} \]  

(4.7)

where \( \theta = \omega + f \) with \( f \) being the true anomaly. The optimal true argument of latitude \( \bar{\theta}_{oop} \) to apply thrust into the normal direction of the RTN-frame in order to reconfigure the out-of-plane motion is given by

\[ \bar{\theta}_{oop} = \tan^{-1} \left( \frac{\Delta \delta i_y}{\Delta \delta i_x} \right) \]  

(4.8)

The feedback gain matrix for eccentric orbits is given by

\[ P = \frac{1}{k} \begin{pmatrix} \cos(J')^N & 0 & 0 & 0 & 0 \\ 0 & \cos(J')^N & 0 & 0 & 0 \\ 0 & 0 & \cos(J')^N & 0 & 0 \\ 0 & 0 & 0 & \cos(H')^N & 0 \\ 0 & 0 & 0 & 0 & \cos(H')^N \end{pmatrix} \]  

(4.9)

with \( J' = \theta - \bar{\theta}_{ip} \) and \( H' = \theta - \bar{\theta}_{oop} \). The user-defined parameters \( k \) and \( N \) are the same as in equation (4.4).
4.2. Controlling the Along-Track Separation Rate

The relative mean longitude (or the along-track separation) $\delta \lambda$ can only be actively controlled via thrust into the radial direction of the RTN frame. Yet, if no radial thrust is to be applied, it can be passively controlled by leveraging natural (Keplerian) orbit dynamics

$$\delta \dot{\lambda} = -\frac{3}{2} n \delta a,$$

where $n$ is the mean motion. The difference in the semi-major axis can be effectively controlled by tangential thrust (see equations (A.7) and (A.8)). Once a difference in the semi-major axes is established, both satellites will drift towards to or away from each other (see figure 4.2). The desired change of $\delta \lambda$ is then only a matter of time and not a matter of thrust any longer. This work assumes desired relative motion that is bounded, that is $\delta a_{\text{ref}} = 0$. The more time constrained a reconfiguration is, the higher should be $\delta a$ in order to achieve a change of the relative mean longitude $\delta \lambda$, because the drift of $\delta \lambda$ is proportional to the established difference in the semi-major axis $\delta a$. The more fuel constrained a reconfiguration is the smaller should be $\delta a$ to reconfigure $\delta \lambda$, because the control effort to be applied is proportional to the desired change $\delta a$. When $\Delta \delta \dot{\lambda}$ becomes zero also $\delta \dot{\lambda}_{\text{ref}}$ should become zero (see figure 4.2). A function that ensures this property and is centered at the desired reference $\delta \lambda_{\text{ref}}$ is given by

$$\delta \dot{\lambda}_{\text{ref}} = \begin{cases} \min \left\{ \frac{|\Delta \delta \dot{\lambda}|}{\tau}, \delta \dot{\lambda}_{\text{des}} \right\}, & \text{if } \Delta \delta \dot{\lambda} \geq 0 \\ -\min \left\{ \frac{|\Delta \delta \dot{\lambda}|}{\tau}, \delta \dot{\lambda}_{\text{des}} \right\}, & \text{if } \Delta \delta \dot{\lambda} < 0 \end{cases},$$

(4.11)

where $\tau \in \mathbb{R}^+$ is an arbitrary large scaling factor, which sets the slope of the set point for $\delta \dot{\lambda}$ (see figure 4.2 (right)). Equation (4.11) is a function of the current error in the along-track separation $\Delta \delta \dot{\lambda}$ and $\delta \dot{\lambda}_{\text{des}}$, which is given by

$$\delta \dot{\lambda}_{\text{des}} = \frac{3}{2} n |\delta a_{\text{des}}|.$$

(4.12)
In order to derive $\delta a_{\text{des}}$ it is important to understand that tangential thrust required for reconfiguring the eccentricity vector also affects $\delta a$. It is not possible to perform a reconfiguration of the eccentricity vector without affecting the difference in semi-major axis. The magnitude of the change of the semi-major axis $\Delta \delta a_{\text{tan}}$ due to a tangential thrust input is given by

$$
\Delta \delta a_{\text{tan}} = \frac{2}{a_c n \sqrt{1 - e^2}} (1 + e \cos \varphi) \cdot \Delta \nu_{\text{tan}},
$$

(4.13)

where the velocity change $\Delta \nu_{\text{tan}} = \Delta \nu_{\text{orbit}} / 2$. The velocity change $\Delta \nu_{\text{orbit}}$ for one orbit is derived in the next section and is given by equation (5.6). The factor of 1/2 is required to only consider one continuous control input that is centered at one of the two fuel-optimal locations for reconfiguring the eccentricity vector (see figure 4.3). Now, the desired change in the semi-major axis is set to

$$
|\delta a_{\text{des}}| = \frac{|\Delta \delta a_{\text{tan}}|}{2}.
$$

(4.14)

Now, the desired drift $\delta \dot{\lambda}_{\text{des}}$ can be determined using equation (4.12). Once $\delta \dot{\lambda}_{\text{ref}}$ is determined, using equation (4.10) it can be translated into a desired difference of the semi-major axes $\delta a$

$$
\delta a = \frac{2 \delta \dot{\lambda}_{\text{ref}}}{3 \, n}.
$$

(4.15)

$\delta a$ is then applied to the controller for tracking and thus the along-track separation $\delta \lambda$ is controlled as well. In order to get the true difference of the semi-major axis $\delta a$ to oscillate about $\delta a_{\text{des}}$ with a magnitude of $\Delta \delta a_{\text{tan}}$ (equation (4.13)) the tracking error $\Delta \delta a = \delta a - \delta a_{\text{des}}$ is limited to

$$
|\Delta \delta a_{\text{max}}| = \frac{|\Delta \delta a_{\text{tan}}|}{2}.
$$

(4.16)

and can be enforced by adjusting the feedback gain matrix $P$ (equation (4.4)) such that the error feedback is set to zero if the maximum error $|\Delta \delta a_{\text{max}}|$ is violated

$$
P|\Delta \delta a_{\text{des}}| \leq |\Delta \delta a_{\text{max}}| \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & \cos(H)^N & 0 \\
0 & 0 & 0 & \cos(H)^N
\end{bmatrix}.
$$

(4.17)

Now, no more tangential thrust is applied that would further destabilize $\delta a$. In this case, thrust will again be applied half an orbit period later when both $\Delta \delta a_{\text{des}}$ and $\Delta \delta e$ can be stabilized at the mean argument of latitude $u = u_{ip}$. The problem is shown in figure 4.3. If tangential thrust stabilizes both the eccentricity vector $\delta e$ and the differential semi-major axis $\delta a$ (location $u_1$ in figure 4.3) the feedback gain matrix $P$ in equation (4.4) is applied. If tangential thrust stabilizes the eccentricity vector $\delta e$ and destabilizes the differential semi-major axis $\delta a$ (location $u_2$ in figure 4.3) and if $|\Delta \delta a_{\text{des}}| \geq |\Delta \delta a_{\text{max}}|$ the feedback gain matrix $P|\Delta \delta a_{\text{des}}| \leq |\Delta \delta a_{\text{max}}|$ in equation (4.17) is applied. Since the differential semi-major axis $\delta a$ is allowed to oscillate around the applied set point $\delta a_{\text{des}}$ by the magnitude $|\Delta \delta a_{\text{max}}|$ also the relative mean longitude (along-track separation) $\delta \lambda$ oscillates. As soon as the eccentricity vector is reconfigured to steady state, also $\delta a$ converges to $\delta a_{\text{ref}} = 0$ and the along-track separation error $\Delta \delta \lambda$ becomes zero without further oscillations.
4.2. Controlling the Along-Track Separation Rate

Figure 4.2: Keplerian drift of the along-track separation $\delta \lambda$ (left side) and control design of the differential semi-major axis $\delta a$ set-point as a function of the along-track separation error $\Delta \delta \lambda$ (right side). The red dashed lines on the right side correspond to different $\tau$ values in equation (4.11). A higher $\tau$ yield a steeper transition.

Figure 4.3: Absolute orbit with fuel optimal locations to apply thrust $u_1$ and $u_2$. Conflict of interests between stabilizing the error in semi-major axis $\Delta \delta a$ (red control inputs, note no direction change - always in along-track or anti-along-track direction for a constant error $\Delta \delta a$) and stabilizing the error in the eccentricity vector $\Delta \delta e$ (green control inputs, note direction change each half orbit period).
5.1. Time constraint

Mission requirements usually impose time constraints on reconfigurations. The proposed control strategy allows to calculate a desired thrust level input that is required in order to achieve a desired $\Delta v$ change. The feedback gain matrix in equation (4.4) defines the shape of the applied control inputs (see figure 5.1). The control inputs are characterized by a cosine shape centered at the fuel-optimal location with the even and positive exponent $N$. The $\Delta v$ change for one orbit period $\Delta v_{2\pi}$ of such continuous control accelerations is given by

$$\Delta v_{2\pi} = u_d \int_0^{2\pi} \cos(u)^N du ,$$  \hspace{1cm} (5.1)

with $u_d$ being the desired control acceleration and $u$ the mean argument of latitude. This integral can be solved using the reduction formula

$$\int \cos(u)^N du = \frac{\sin(u) \cos(u)^{N-1}}{N} + \frac{N-1}{N} \int \cos(u)^{N-2} du .$$  \hspace{1cm} (5.2)

Figure 5.1: Control inputs of form $u_d \cos(u)^N$. Hatched area represents delta-$v$ change of one orbit period, $u_d$ is the desired control input.
Note that $N$ is an even number. Therefore, the term $\sin(u)\cos(u)^{N-1}$, with an odd exponent for the cosine, can always be expressed in terms of $\sin(k\cdot u)$ with $k$ being an even number. An example for $N = 3$ is

$$
\frac{1}{4} \sin(u)\cos(u)^3 = \frac{1}{4} \left[ \sin(u)\cos(u) \cdot \cos(u)^2 \right],
$$

$$
= \frac{1}{4} \left[ \frac{1}{2} \sin(2u) \cdot \left( \frac{1}{2} + \frac{1}{2} \cos(2u) \right) \right],
$$

$$
= \frac{1}{16} \sin(2u) + \frac{1}{16} \sin(2u)\cos(2u),
$$

$$
= \frac{1}{16} \sin(2u) + \frac{1}{32} \sin(4u).
$$

(5.3)

For the integration constants $0$ to $2\pi$ these expression (5.3) becomes zero. Since $\sin(u)\cos(u)^{N-1}$ only has odd exponents, the first summand of equation (5.2) is always zero and can be neglected. Equation (5.2) is rewritten as

$$
\int_0^{2\pi} \cos(u)^N du = \frac{N-1}{N} \int_0^{2\pi} \cos(u)^{N-2} du.
$$

(5.4)

Now, one can easily see that this reduction formula can be applied sequentially until the exponent $N-2$ is "reduced" to $N-2 = 2$, which is given for $N = 4$. This product of a sequence is given by

$$
\Delta v_{2\pi} = u_d \prod_{i=N,N-2,\ldots}^{i=4} \left[ \frac{i-1}{i} \right] \cdot \int_0^{2\pi} \cos(u)^2 du
$$

| which can be rewritten as

$$
= u_d \prod_{i=N,N-2,\ldots}^{i=4} \left[ \frac{i-1}{i} \right] \cdot \int_0^{2\pi} \frac{1}{2} + \frac{1}{2} \cos(2u) du
$$

| substitute $x = 2u$ and $\frac{dx}{du} = 2$

$$
= u_d \prod_{i=N,N-2,\ldots}^{i=4} \left[ \frac{i-1}{i} \right] \cdot \left( \frac{1}{2} \int_0^{2\pi} \cos(x) dx \right)_{0}^{2\pi}
$$

| adjust integration boundaries

$$
= u_d \prod_{i=N,N-2,\ldots}^{i=4} \left[ \frac{i-1}{i} \right] \cdot \left( \pi + \frac{1}{4} \sin(x) \right)_{0}^{2\pi}
$$

| resubstitute $x = 2u$

$$
= u_d \prod_{i=N,N-2,\ldots}^{i=4} \left[ \frac{i-1}{i} \right] \cdot \left( \pi + \frac{1}{4} \sin(2u) \right)_{0}^{2\pi}
$$

| simplify

$$
\Delta v_{2\pi} = u_d \pi \prod_{i=N,N-2,\ldots}^{i=4} \left[ \frac{i-1}{i} \right] \left[ \left( \frac{m}{s^2} \cdot \text{rad}/\text{orbit} \right) \right].
$$

(5.5)

In order to obtain the desired units a conversion into the time domain has to be performed

$$
\Delta v_{\text{orbit}} = \Delta v_{2\pi} \cdot \frac{T}{2\pi} \left[ \left( \frac{m}{s} \right)/\text{orbit} \right],
$$

(5.6)

where $T$ is the chief’s orbit period. For a desired reconfiguration with a required optimal total velocity change $\Delta v_{i,p}^{\text{opt}}$ for the in-plane reconfiguration and $\Delta v_{o,oop}^{\text{opt}}$ for the out-of-plane
reconfiguration are given by

\[ \Delta v_{ip}^{\text{opt}} = a_c n \cdot \frac{\| \Delta \delta e \|}{2\eta} \quad \text{see [5],} \quad (5.7) \]

\[ \Delta v_{oop}^{\text{opt}} = a_c n \frac{1 - e_c}{\eta} \| \Delta \delta i \| , \quad (5.8) \]

with \( \eta = \sqrt{1 - e_c^2} \). See the derivation of equation (5.8) in appendix B. In order for equation (5.7) to hold true it is important that \( \frac{\Delta \delta a}{2(1 + e_c)} \) is smaller than \( \frac{1}{2\eta^2} \). For reconfigurations from one bounded formation to another bounded formation this restriction is always satisfied. Since the proposed control strategy can only apply thrust into either the tangential direction or the normal direction for a reconfiguration the number of orbits after which the reconfiguration shall be completed has to be split up into a part for the in-plane and the out-of-plane reconfiguration

\[ \# \text{orbits}_{ip} = \frac{\| \Delta \delta e \|}{2(1 - e_c)\| \Delta \delta i \| + \| \Delta \delta e \|} \cdot \# \text{orbits} , \quad (5.9) \]

\[ \# \text{orbits}_{oop} = \frac{2(1 - e_c)\| \Delta \delta i \| + \| \Delta \delta e \|}{2(1 - e_c)\| \Delta \delta i \| + \| \Delta \delta e \|} \cdot \# \text{orbits} . \quad (5.10) \]

By defining \( \Delta v_{2\pi} = u_d \cdot \Delta \dot{v}_{2\pi} \) the velocity change that is required during one orbit period is given by

\[ \frac{\Delta v^{\text{opt}}}{\# \text{orbits}} = u_d \Delta \dot{v}_{2\pi} \cdot \frac{T}{2\pi} \left[ \frac{m}{s} \right] / \text{orbit} , \quad (5.11) \]

with the only unknown being \( u_d \). Solving for \( u_d \) yields

\[ u_d = \Delta v^{\text{opt}} \cdot \frac{2\pi}{\# \text{orbits}} \left[ \frac{m}{s^2} \right] , \]

\[ u_d = \frac{\Delta v^{\text{opt}} \cdot 2\pi}{\pi \cdot \# \text{orbits}} \prod_{i=N,N-2,...}^{4} \frac{i}{i-1} \left[ \frac{m}{s^2} \right] , \]

\[ u_d = \frac{2\Delta v^{\text{opt}}}{T \cdot \# \text{orbits}} \prod_{i=N,N-2,...}^{4} \frac{i}{i-1} \left[ \frac{m}{s^2} \right] . \quad (5.12) \]

The desired thrust levels into the normal \( u_{d,oop} \) and the tangential direction \( u_{d,ip} \) are therefore given by

\[ u_{d,ip} = \frac{2\Delta v_{ip}^{\text{opt}}}{T \cdot \# \text{orbits}_{ip}} \prod_{i=N,N-2,...}^{4} \frac{i}{i-1} \left[ \frac{m}{s^2} \right] , \quad (5.13) \]

\[ u_{d,oop} = \frac{2\Delta v_{oop}^{\text{opt}}}{T \cdot \# \text{orbits}_{oop}} \prod_{i=N,N-2,...}^{4} \frac{i}{i-1} \left[ \frac{m}{s^2} \right] . \quad (5.14) \]

Note to ensure that the thrusters are capable of producing the desired control acceleration \( u_d \). Furthermore, since the along-track separation is only passively controlled by leveraging Keplerian dynamics as described in section 4.2, the maximum error in the along-track direction \( |\Delta \delta \lambda| \) has to be smaller than

\[ |\Delta \delta \lambda| \leq \frac{3}{2} n |\delta a_a| \cdot \# \text{orbits} \cdot T . \quad (5.15) \]

For the definition of \( \delta a_a \) see equation (4.15).
5.2. Thrust Constraint

The NetSat cubesats are equipped with a low-thrust level propulsion system - nano field emission electrical propulsion (NanoFEEP) - with a maximum combined thrust level of $80\,\mu N$ [3]. It is important that the thrusters are not operated in saturation. Therefore a thrust constraint is formulated that ensures that a user defined thrust level is not exceeded. Note, that the time constraint and the thrust constraint are mutually exclusive. The thrust constraint is given by

$$\| F \| \leq F_{\text{desired}}$$

where $F_{\text{desired}}$ has to be within the thruster levels, but larger than the relative dynamic perturbations such that a reconfiguration or formation maintenance is still possible.

5.3. Circular No-Entry-Zone Constraint

Define a circular boundary of radius $r_i \in \mathbb{R}$, centered at $x_i \in \mathbb{R}^2$ for any combination of two ROEs

$$\| G_i^T \mathbf{d}a - x_i \|_2 \geq r_i$$

with $G_i \in \mathbb{R}^{2 \times 5}$ and $r_i > 0$. $G_i$ defines the ROE vector elements to be constrained with all elements being zero but two elements which corresponds to the constrained elements is 1. Example: For constraining $\pm e$ one has to define

$$G = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}^T,$$

such that $G^T \mathbf{d}a = \mathbf{d}e$.

5.4. Passive Collision Avoidance

For satellite formations it is convenient to acquire passive collision safety by defining a safety distance $\varepsilon$ in the rn-plane of the RTN frame. The minimum relative distance for bounded motion ($\delta a = 0$) in the rn-plane throughout one orbit period is given by [6]

$$\delta r_{\text{rn}}^{\text{min}} = \frac{\sqrt{2}|\mathbf{d}e \cdot \delta i|}{(\delta e^2 + \delta i^2 + |\mathbf{d}e + \delta i| \cdot |\mathbf{d}e - \delta i|)^{1/2}} \geq \varepsilon,$$

which is applicable for near-circular orbits and for $\delta a = 0$.

5.5. Wall Constraint

Wall constraints describe limits for state-space variables that restrict the corresponding state to enter these restricted areas. Those constraints can be formulated using [13]:

$$c_i^T \mathbf{d}a \leq d_i,$$

$$c_i^T \mathbf{d}a \geq d_i,$$

with $c_i \in \mathbb{N}^{1 \times 6}$ and $d_i \in \mathbb{R}^+$. $c_i$ defines the ROE vector element to be constrained with all elements being zero but one element which corresponds to the constrained element is 1. An example for constraining $\delta i_x$ one has to define

$$c = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T,$$

such that $c^T \mathbf{d}a = \delta i_x$. 
Guidance Strategy - Reference Governor

This section deals with the guidance of the applied reference \( \delta \mathbf{a}_a \), at which the Lyapunov controller stabilizes the relative state, in order to satisfy the constraints imposed on the re-configuration. A reference governor (RG) that will guide the applied reference \( \delta \mathbf{a}_a \) through the ROE state-space while ensuring the satisfaction of the constraint set is developed. The RG uses an attractive potential field that is directed to the desired reference \( \delta \mathbf{a}_r \) and the constraints are realized by repulsive potential functions. Together both the repulsive and the attractive fields compose a potential field map \( \varphi \) with gradient \( \mathbf{p} \) at current applied reference \( \delta \mathbf{a}_a \). Thus, the ROE state is guided along this gradient to the desired state without violating any constraints. Following the potential field gradient \( \mathbf{p} \) ensures that the steady state error is minimum without violating the constraint set (see figure 6.1). The rate of change at which the applied reference is guided \( |\dot{\delta \mathbf{a}_a}| \), is controlled by formulating a Lyapunov threshold \( \Gamma \) for the Lyapunov function given by equation 4.2. The Lyapunov threshold represents a maximum allowed tracking error since the Lyapunov value \( V = 0.5 \Delta \delta \mathbf{a}_a \Delta \delta \mathbf{a}_a \) is a function of the tracking error itself. A Lyapunov threshold \( \Gamma_i \) is derived for each constraint that is imposed on the relative satellite state and the most restrictive Lyapunov threshold

\[
\Gamma = \min \{ \Gamma_i \} \quad (6.1)
\]

is to be enforced by the RG [12]. Note that the derived Lyapunov thresholds \( \Gamma_i \) are suited for the specific Lyapunov function in use and change if a different Lyapunov function is used. All \( \Gamma_i \) are derived in the following sections. If the Lyapunov value \( V \) is smaller than its threshold \( \Gamma \) all constraints are satisfied. Thus all constraints are translated into a single constraint

\[
V \leq \Gamma \quad , \quad (6.2)
\]

which can be enforced using [13]

\[
\delta \dot{\mathbf{a}}_a = \kappa \left[ \Gamma - V \right] \cdot \mathbf{p} \quad , \quad (6.3)
\]

where \( \kappa \in \mathbb{R}^+ \) is an arbitrary large scalar. Equation (6.3) is the central equation of the RG. If the Lyapunov value \( V \) gets close to the Lyapunov threshold, the change of the applied reference tends to zero, because of \( \lim_{V \to \Gamma} \left[ \Gamma - V \right] = 0 \), giving the system time to converge to the applied reference. If the ROE state is exactly at the applied reference \( V = 0 \) the change
of the applied reference is maximized such that $\delta \alpha_a$ reaches $\delta \alpha_r$ as quickly as possible. The vector field is given by [12]

$$\mathbf{p} = \nabla \phi \ .$$

The operator $\nabla(.)$ denotes taking the partial derivatives with respect to the applied reference $\delta \alpha_a$:

$$\nabla = \left( \frac{\partial}{\partial \delta a}, \frac{\partial}{\partial \delta e_x}, \frac{\partial}{\partial \delta e_y}, \frac{\partial}{\partial \delta i_x}, \frac{\partial}{\partial \delta i_y} \right)_a \ .$$

The vector field is a conservative function that tends to zero at $\delta \alpha_a = \delta \alpha_r$ and goes to infinity at the constraint boundaries. Since the sum of conservative functions is also conservative, $\mathbf{p}$ can be constructed by taking into account separate terms for each defined constraint [12]. After all potential field gradients are found the vector field $\mathbf{p}$ is given by

$$\mathbf{p} = -\nabla \tilde{\phi} - \sum_{i=1}^{N} \nabla \phi_i \ ,$$

where $\tilde{\phi}$ is the global attractive potential field (see section 6.1) and $\phi_i$ are the repulsive potential fields of the constraints. A potential function for the constraints is given by [12]

$$\phi_i = \begin{cases} 
-\frac{Y_i^2(C_i - C_i)}{(C_i - C_i)C_i}, & \text{if } C_i \leq \zeta_i \\
0, & \text{otherwise}
\end{cases} \ ,$$

where $C_i$ is the margin of the constraint $i$, which is to be enforced, and with an arbitrary small safety margin $Y_i \in \mathbb{R}^+$ and influence distance $\zeta_i \in \mathbb{R}^+$, where $\zeta_i > Y_i$, $\forall i = 1, ..., N$. The gradient of this potential function is derived as
Figure 6.2: Plot of the global potential field and gradient magnitudes for both the global attractive field (left) and the constraints (right). Note that the gradient magnitude for the global attractive field is unity for $\|\delta \alpha_a - \delta \alpha_r\| \geq \eta$. Furthermore, as soon as the relative state approached a constraint’s safety margin $\gamma$ the constraints gradient magnitude becomes $-1$ and is able to negate the global potential field (equilibrium, the constraint cannot be violated). The Local repulsive fields are only considered if the distance to the constraint is less than the influence region $\xi$ (dotted line parts do not contribute).

\[\nabla \phi_i = \frac{-2Y_i^2(\zeta_i - C_i)\nabla C_i \cdot (\zeta_i^2 - Y_i^2)C_i - Y_i^2(\zeta_i - C_i)^2 \cdot (\zeta_i^2 - Y_i^2)\nabla C_i}{(\zeta_i^2 - Y_i^2) C_i^2},\]
\[= \frac{Y_i^2(\zeta_i - C_i)\nabla C_i [-2C_i - (\zeta_i - C_i)]}{(\zeta_i^2 - Y_i^2) C_i^2},\]
\[= \frac{-Y_i^2(\zeta_i - C_i)(\zeta_i + C_i)}{(\zeta_i^2 - Y_i^2) C_i} \nabla C_i,\]
\[= \frac{-Y_i^2(\zeta_i^2 - C_i^2)}{(\zeta_i^2 - Y_i^2) C_i^2} \nabla C_i.\] (6.7)

Equation (6.7) is derived as a function of $\nabla C_i$, which is referred to as the directional parameter from here on. It provides the directional gradient information of the potential field and is derived for each constraint in the following sections.

### 6.1. The Global Potential Field and its Gradient

The global attractive potential function is given by

\[\tilde{\phi} = \begin{cases} ||\delta \alpha_a - \delta \alpha_r||, & \text{if } ||\delta \alpha_a - \delta \alpha_r|| \geq \eta \\ \frac{1}{2} ||\delta \alpha_a - \delta \alpha_r||^2 + \frac{1}{2} \eta^2, & \text{otherwise} \end{cases} \] (6.8)

where $\eta \geq 1$ and $\eta \in \mathbb{R}$ is an arbitrary small user defined number [12]. The motivation for the two conditions and the definition of $\eta$ is shown in figure 6.3. The second condition in equation (6.8) is important such that the directional derivative of the global potential field (the vector field which determines the change of the applied reference $\delta \tilde{\alpha}_a$ becomes zero
for $\delta \mathbf{a}_a = \delta \mathbf{a}_r$. In the following a step-by-step derivation of the gradient of $\phi$ is shown. For the case of $||\delta \mathbf{a}_a - \delta \mathbf{a}_r|| < \eta$ the gradient is

$$
\nabla \phi = \nabla \left[ \frac{1}{2\eta} \left( (\delta a_a - \delta a_r)^2 + (\delta e_{xa} - \delta e_{xr})^2 + (\delta e_{ya} - \delta e_{yr})^2 + (\delta i_{xa} - \delta i_{xr})^2 + (\delta i_{ya} - \delta i_{yr})^2 \right) + \frac{1}{2\eta} \right],
$$

$$
= \frac{1}{\eta} \left[ (\delta a_a - \delta a_r) \frac{\partial}{\partial \delta a_a} + (\delta e_{xa} - \delta e_{xr}) \frac{\partial}{\partial \delta e_{xa}} + (\delta e_{ya} - \delta e_{yr}) \frac{\partial}{\partial \delta e_{ya}} + (\delta i_{xa} - \delta i_{xr}) \frac{\partial}{\partial \delta i_{xa}} + (\delta i_{ya} - \delta i_{yr}) \frac{\partial}{\partial \delta i_{ya}} \right].
$$

$$
= \frac{\delta \mathbf{a}_a - \delta \mathbf{a}_r}{\eta}. \quad (6.9)
$$

For the case of $||\delta \mathbf{a}_a - \delta \mathbf{a}_r|| \geq \eta$ the gradient of the global potential field is derived as

$$
\nabla \phi = \nabla \left[ (\delta a_a - \delta a_r)^2 + (\delta e_{xa} - \delta e_{xr})^2 + (\delta e_{ya} - \delta e_{yr})^2 + (\delta i_{xa} - \delta i_{xr})^2 + (\delta i_{ya} - \delta i_{yr})^2 \right]^{0.5} = \nabla \sqrt{W},
$$

$$
= \frac{\partial}{\partial \delta a_a} + \frac{\partial}{\partial \delta e_{xa}} + \frac{\partial}{\partial \delta e_{ya}} + \frac{\partial}{\partial \delta i_{xa}} + \frac{\partial}{\partial \delta i_{ya}}.
$$

$$
= \frac{1}{2} W^{-0.5} \cdot 2(\delta a_a - \delta a_r) \frac{\partial}{\partial \delta a_a} + \frac{1}{2} W^{-0.5} \cdot 2(\delta e_{xa} - \delta e_{xr}) \frac{\partial}{\partial \delta e_{xa}} + \frac{1}{2} W^{-0.5} \cdot 2(\delta e_{ya} - \delta e_{yr}) \frac{\partial}{\partial \delta e_{ya}} + \frac{1}{2} W^{-0.5} \cdot 2(\delta i_{xa} - \delta i_{xr}) \frac{\partial}{\partial \delta i_{xa}} + \frac{1}{2} W^{-0.5} \cdot 2(\delta i_{ya} - \delta i_{yr}) \frac{\partial}{\partial \delta i_{ya}}.
$$

$$
= \frac{1}{\sqrt{W}} \left[ (\delta a_a - \delta a_r) \frac{\partial}{\partial \delta a_a} + (\delta e_{xa} - \delta e_{xr}) \frac{\partial}{\partial \delta e_{xa}} + (\delta e_{ya} - \delta e_{yr}) \frac{\partial}{\partial \delta e_{ya}} + (\delta i_{xa} - \delta i_{xr}) \frac{\partial}{\partial \delta i_{xa}} + (\delta i_{ya} - \delta i_{yr}) \frac{\partial}{\partial \delta i_{ya}} \right].
$$

$$
= \frac{\delta \mathbf{a}_a - \delta \mathbf{a}_r}{||\delta \mathbf{a}_a - \delta \mathbf{a}_r||}. \quad (6.10)
$$
Thus, the gradient can be given in a single equation by

$$\nabla \dot{\phi} (\delta \alpha_a, \delta \alpha_r) = \frac{\delta \alpha_a - \delta \alpha_r}{\max(||\delta \alpha_a - \delta \alpha_r||, \eta)},$$

(6.11)

which is unitary if ||\delta \alpha_a - \delta \alpha_r|| \geq \eta and tends to zero for ||\delta \alpha_a - \delta \alpha_r|| < \eta.

### 6.2. Lyapunov Threshold for the Time Constraint

The time constraint can be enforced without a potential field and only a Lyapunov threshold \( \Gamma_{\text{thrust}} \). The applied control thrust is proportional to the tracking error \( \Delta \delta \alpha \) (see the control law in equation 4.1). Based on the current tracking error direction \( \hat{\rho}_e = \frac{\Delta \delta \alpha}{\Delta \delta \alpha} \), and assuming that this error direction is constant one can calculate the control input \( \hat{u}_c \) that is going to be applied by the controller per unit error in this direction

$$\hat{u}_c = \| B^* P \hat{\rho}_e \| .$$

(6.12)

The error into the current tracking error direction that is required in order to yield the desired control acceleration \( u_d \) can therefore be determined using equation (6.12) and is given by

$$\| \Delta \delta \alpha_{\text{req}} \| = \frac{u_d}{\hat{u}_c} ,$$

(6.13)

which can easily be transferred into the Lyapunov form of equation 4.2 to yield the Lyapunov threshold for the maximum thrust constraint by

$$\Gamma_{\text{time}} = \frac{1}{2} \| \Delta \delta \alpha_{\text{req}} \|^2 ,$$

$$= \frac{1}{2} \left( \frac{u_d}{\| B^* P \hat{\rho}_e \|} \right)^2 .$$

(6.14)

The inverse of the control input matrix \( B^* \) and the feedback gain matrix \( P \) are calculated at the fuel-optimal mean arguments of latitude for in-plane reconfigurations \( \hat{u}_{ip} \) and for out-of-plane reconfigurations \( \hat{u}_{oop} \).

### 6.3. Lyapunov Threshold for the Thrust Constraint

The Lyapunov threshold for the thrust constraint is derived the same way as the Lyapunov threshold for the time constraint (see equation (6.15)). The only difference that the desired control input is no longer a function of time, but a constant user defined value

$$\Gamma_{\text{thrust}} = \frac{1}{2} \left( \frac{F_{\text{desired}}/m_{s/c}}{\| B^* P \hat{\rho}_e \|} \right)^2 ,$$

(6.15)

with \( m_{s/c} \) being the spacecraft mass. It is important to note, that either the time or the thrust constraint can be applied at a time. It is not possible to enforce both the time and the thrust constraint at the same time.
6.4. Wall Constraint

A wall constraint is defined as one of the 5 ROE vector elements to be constrained to be always larger or always smaller than a defined value $d$. The constrained ROE vector element is defined by $c_i \in \mathbb{N}^5$ with all elements being zero but one element which corresponds to the constrained element is 1 (e.g. for constraining $\delta i_y$ one has to define $c = (0,0,0,0,1)^T$, thus $c^T \delta \alpha = \delta i_y$).

6.4.1. Lyapunov Threshold

The margin of the constrained element of $\delta \alpha_a$ is given by

$$C_i = |c_i^T \delta \alpha_a - d_i| , \quad (6.16)$$

where $d_i$ is the boundary of the wall constraint (see equation 5.20). Now the error of the ROE vector element should not become larger than this margin. But since the Lyapunov function $V$ is of scalar nature one cannot determine into which direction the current tracking error $\Delta \delta \alpha$ is based just on the knowledge of $V$. Only information of the tracking error size is provided. Therefore, all components of the tracking error vector $\Delta \delta \alpha$ contribute to each defined constraint, even though if not all components are constrained. This makes all constraints more restrictive. The Lyapunov threshold is found by rewriting the margin in the Lyapunov function form $V = \frac{1}{2} \cdot \Delta \delta \alpha^2$. The result for wall constraints is given by

$$\Gamma_i(\delta \alpha_a) = \frac{1}{2} C_i^2 ,$$

$$\Gamma_i(\delta \alpha_a) = \frac{1}{2} (c_i^T \delta \alpha_a - d_i)^2 . \quad (6.17)$$

6.4.2. Gradient of the Potential Field

For the wall constraint of form (5.20) $c_i \delta \alpha \leq d_i$ the directional parameter $\nabla C_i$ is calculated as

$$\nabla C_i = \nabla (d_i - c_i^T \delta \alpha_a) ,$$

$$= -c_i^T \frac{\partial \delta \alpha_a}{\partial \delta \alpha_a} - c_i^T \frac{\partial \delta \alpha_a}{\partial \delta e_xa} - c_i^T \frac{\partial \delta \alpha_a}{\partial \delta e_ya} - c_i^T \frac{\partial \delta \alpha_a}{\partial \delta i_xa} - c_i^T \frac{\partial \delta \alpha_a}{\partial \delta i_ya} ,$$

$$= -c_i^T \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} - c_i \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} - c_i \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} - c_i \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} - c_i \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} ,$$

$$= -c_i . \quad (6.18)$$

Thus, the gradient of the potential field of form (6.7) for this constraint is

$$\nabla \phi_i = \frac{-Y_i^2 (c_i^2 - d_i^2)}{(c_i^2 - Y_i^2) C_i^2} \nabla C_i ,$$

$$\nabla \phi_i = \frac{Y_i^2 (c_i^2 - d_i^2)}{(c_i^2 - Y_i^2) C_i^2} c_i . \quad (6.19)$$
For the wall constraint of form (5.21) the directional parameter $\nabla C_i$ is calculated as

$$
\nabla C_i = \nabla (-d_i + c_i^T \delta \alpha_a),
$$

$$
= + c_i^T \frac{\partial \delta \alpha_a}{\partial a_a} + c_i^T \frac{\partial \delta \alpha_a}{\partial e_{xa}} + c_i^T \frac{\partial \delta \alpha_a}{\partial e_{ya}} + c_i^T \frac{\partial \delta \alpha_a}{\partial l_{xa}} + c_i^T \frac{\partial \delta \alpha_a}{\partial l_{ya}},
$$

$$
= + c_i^T \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + c_i^T \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + c_i^T \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + c_i^T \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}.
$$

Thus, the gradient of the potential field of form (6.7) is given by

$$
\nabla \phi_i = -Y_i^2(\zeta_i^2 - C_i^2) \nabla C_i,
$$

$$
\nabla \phi_i = \frac{-Y_i^2(\zeta_i^2 - C_i^2)}{(\zeta_i^2 - Y_i^2)C_i^2} c_i. \tag{6.21}
$$

The only difference to equation (6.19) is a change of sign.

**6.5. Circular No-Entry-Zone Constraint**

The circular no-entry-zone constraint can be either applied on the eccentricity vector or the inclination vector, or arbitrary combinations of two of their elements. It is not possible to define such a constraint for $\delta a$ due to the necessity to accurately track $\delta a_{ref}$ in order to be able to control relative mean longitude $\delta \lambda$. Remember, the ROE vector elements to be constrained are defined by $G_i \in \mathbb{R}^{2 \times 5}$ with all elements being zero but two elements which corresponds to the constrained elements is 1 (e.g. for constraining $\delta e$ one has to define

$$
G = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}^T,
$$

such that $G^T \delta \alpha = \delta e$).

**6.5.1. Lyapunov Threshold**

The margin of the constrained elements of $\delta \alpha_a$ is given by

$$
C_i = \| G_i^T \delta \alpha_a - x_i \| - r_i, \tag{6.23}
$$

where $x_i$ and $r_i$ are the center and radius of the no-entry-zone respectively. In analogy to equation (6.17) the Lyapunov threshold for constraint 5.17 can be expressed by

$$
\Gamma_i = \frac{1}{2} C_i^2,
$$

$$
\Gamma_i = \frac{1}{2} (\| G_i^T \delta \alpha_a - x_i \| - r_i)^2. \tag{6.24}
$$
6.5.2. Gradient of the Potential Field

The directional parameter $\nabla C_i$ is calculated as

$$\nabla C_i = \nabla \left( \| G_i^T \delta a - x_i \| - r_i \right),$$

$$= \nabla \left[ \left( (G^T \delta a(1) - x(1))^2 + (G^T \delta a(2) - x(2))^2 \right)^{0.5} - r_i \right],$$

$$= \nabla \left[ \sqrt{W} - r_i \right],$$

$$= \frac{1}{2\sqrt{W}} \left( \frac{\partial W}{\partial \delta a_a} \frac{\partial W}{\partial \delta e_{x\alpha}} \frac{\partial W}{\partial \delta e_{y\beta}} \frac{\partial W}{\partial \delta e_{t\alpha}} \frac{\partial W}{\partial \delta t_{\beta}} \right)^T,$$

the outer derivative of $W$ is of form $2(G_i^T \delta a - x_i)$ and the inner derivative is 1 if it is the partial derivative of one of the constrained ROE vector elements. By multiplying with $G_i$ all partial derivatives of the non-constrained ROE vector elements become zero, as mathematically correct,

$$= \frac{G_i \cdot 2(G_i^T \delta a - x_i)}{2\sqrt{W}},$$

$$= \frac{G_i^T \delta a - x_i}{\| G_i^T \delta a - x_i \|}.$$  \hspace{1cm} (6.26)

Thus, the gradient of the potential field of form (6.7) is given by

$$\nabla \phi_i = \frac{-Y_i \left( \frac{\partial^2}{\partial \delta e_{x\alpha}} \dot{\delta a}_r \right) C_i^2}{\left( \dot{\delta a}_r \right)^2 C_i^2} \nabla C_i,$$

$$\nabla \phi_i = \frac{-Y_i \left( \frac{\partial^2}{\partial \delta e_{y\beta}} \dot{\delta a}_r \right) C_i^2}{\left( \dot{\delta a}_r \right)^2 C_i^2} \left( G_i^T \delta a - x_i \right),$$

$$\nabla \phi_i = \frac{-Y_i \left( \frac{\partial^2}{\partial \delta e_{t\alpha}} \dot{\delta a}_r \right) C_i^2}{\left( \dot{\delta a}_r \right)^2 C_i^2} \left( G_i^T \delta a - x_i \right).$$  \hspace{1cm} (6.27)

6.6. Passive Collision Avoidance Constraint

6.6.1. Lyapunov Threshold

For small integration time steps the tracking error direction $\frac{\Delta \delta a}{\delta a_r}$ can be assumed to be constant. Based on the difference between the minimum distance in the $rn$-plane of the actual relative state and the defined minimum cross-track separation $\delta r_{rn}^\text{min} (\delta a) - r_i$, the maximum allowable tracking error $\Delta \delta a_{\text{max}}$ can be determined by performing a first order Taylor expansion of equation 5.19 at the actual relative state $\delta a$ (see equations (A.9) - (A.12)). The maximum allowable separation is given by

$$\Delta \delta a_{\text{max}} = \Delta \delta a \left( 1 - \left[ \frac{\partial r_{rn}^\text{min}}{\partial \delta e_{x\alpha}} \frac{\partial r_{rn}^\text{min}}{\partial \delta e_{y\beta}} \frac{\partial r_{rn}^\text{min}}{\partial \delta e_{t\alpha}} \frac{\partial r_{rn}^\text{min}}{\partial \delta t_{\beta}} \right] \delta a \cdot \Delta \delta a_{a_r} \right).$$  \hspace{1cm} (6.28)

The numerator is the current margin of the actual separation and the allowed minimum separation. The denominator describes the unit change of $\delta r_{rn}^\text{min}$ if the error is increased into the current tracking error direction (assumption of constant tracking error direction). Therefore the fractional expression of equation (6.28) expresses how much more the current tracking error is allowed to become larger. Due to the negative nature of this fraction
the term given in brackets will be greater than 1 and the solution is the maximum error that is allowed for the constraint to be satisfied. Note that for the passive collision avoidance constraint only tracking errors of the eccentricity vector $\delta e$ and the inclination vector $\delta i$ are of concern (for this constraint: $\delta \alpha = [\delta e^T \delta i^T]^T$). Furthermore, it is noteworthy that a tracking error not always results in a smaller minimum distance in the $rn$-plane. In fact, an increase of the tracking error in the current tracking error direction can increase the minimum cross-track separation. Therefore, only if the change of $\delta r_{rn}^{\min}$ is negative based on the current tracking error, equation (6.28) is applicable. Thus, the Lyapunov constraint $\Gamma_{rn}$ is determined by

$$
\Gamma_{rn} = \begin{cases} 
\frac{1}{2} \Delta \delta \alpha_{\text{max}}^T \Delta \delta \alpha_{\text{max}}, & \text{for } \left[ \frac{\partial \delta r_{rn}^{\min}}{\partial \delta e_x} \frac{\partial \delta r_{rn}^{\min}}{\partial \delta e_y} \frac{\partial \delta r_{rn}^{\min}}{\partial \delta i_x} \frac{\partial \delta r_{rn}^{\min}}{\partial \delta i_y} \right] \delta \alpha \cdot \Delta \delta \alpha_{a} < 0 \\
\infty, & \text{elsewhere}
\end{cases} 
$$

(6.29)

6.6.2. Potential Field

In the presence of uncertainties of in the along-track separation $\delta \lambda$ it is convenient to achieve passive collision avoidance by establishing a minimum separation in the $rn$-plane of the RTN frame. Such a minimum separation is geometrically represented by a circle with radius $e$ in the $rn$-plane and mathematically formulated by constraint 5.19. Constraint 5.19 can be reformulated by the following three expressions [10]

$$
\| \delta e \| \geq e, \quad \| \delta i \| \geq e, \quad \| \delta e \cdot \delta i \| \geq e\sqrt{\delta e^2 + \delta i^2 - \epsilon^2}.
$$

(6.30) (6.31) (6.32)

The argument of the passive collision avoidance constraint is chosen to be the inequality with the least margin to its boundary

$$
C_{rn} = \min \left\{ \frac{\| \delta e \| - \epsilon}{\| \delta e \| - \epsilon}, \frac{\| \delta i \| - \epsilon}{\| \delta i \| - \epsilon}, \frac{\| \delta e \cdot \delta i \| - \epsilon\sqrt{\delta e^2 + \delta i^2 - \epsilon^2}}{\| \delta e \cdot \delta i \| - \epsilon\sqrt{\delta e^2 + \delta i^2 - \epsilon^2}} \right\}.
$$

(6.33)

The idea is to negate the global potential field (for the eccentricity and the inclination vector) if one of these constraints is approaching zero and to establish collision safety in the $rt$-plane instead of having collision safety in the $rn$-plane (which is about to be violated due to a reconfiguration). If collision safety is established in the $rt$-plane the reconfiguration can proceed. Collision safety in the $rt$-plane is provided by satisfying the following constraint [10]

$$
|\delta \lambda| \geq 2 \cdot \| \delta e \| + \epsilon
$$

(6.34)

and is visualized in figure 6.4. Now, if the relative state has to be guided through an area that violates $rn$-plane safety, the global potential field has to be negated and the reference for the along-track separation $\delta \lambda$ has to be updated such that equation (6.34) is satisfied. The potential field for the $rn$-plane collision safety is constructed the same way as the potential field for the locally constrained state-space areas (equation (6.6)), including a safety margin
Figure 6.4: Collision safety guaranteed in the rt-plane. Left: Safe along track separation for a given eccentricity vector. Right: Relative motion projected in the rt-plane[10].

\[ Y_{rn} \] and an influence region \( \zeta_{rn} \) and is given by

\[
\phi_{rn} = \begin{cases} 
\frac{-Y_{rn}^2 (\zeta_{rn} - C_{rn})^2}{(\zeta_{rn}^2 - Y_{rn}^2) C_{rn}}, & \text{if } C_{rn} \leq \zeta_{rn} \text{ and } |\delta \lambda| < 2 \cdot \|\delta e\| + \varepsilon \\
0, & \text{otherwise}
\end{cases} \tag{6.35}
\]

The gradient of \( \phi_{rn} \) shall negate the gradient of the global potential field \( \nabla \hat{\phi} \) at the safety margin which is given by

\[
\nabla \phi_{rn} = \phi_{rn} \cdot (-\nabla \hat{\phi}) \tag{6.36}
\]

Note, that \( \phi_{rn} \) has a magnitude of 1 at the safety margin. If collision safety has to be established in the rt-plane, the safe along-track separation \( \delta \lambda_{safe} \) is given by

\[
\delta \lambda_{safe} = \begin{cases} 
2 \cdot \|\delta e\| + \varepsilon, & \text{if } C_{rn} \leq \zeta_{rn} \text{ and } |\delta \lambda| < 2 \cdot \|\delta e\| + \varepsilon \text{ and } \delta \lambda > 0 \\
-2 \cdot \|\delta e\| - \varepsilon, & \text{if } C_{rn} \leq \zeta_{rn} \text{ and } |\delta \lambda| < 2 \cdot \|\delta e\| + \varepsilon \text{ and } \delta \lambda \leq 0
\end{cases} \tag{6.37}
\]

which distinguishes the two cases \( \delta \lambda > 0 \) and \( \delta \lambda \leq 0 \). As soon as the constraint margin becomes smaller than the influence region \( (C_{rn} \leq \zeta_{rn}) \) the desired along-track separation \( \delta \lambda_{ref} \) has to be redefined to

\[
\delta \lambda_{ref} = \delta \lambda_{safe} \tag{6.38}
\]

until passive collision safety is established again \( (C_{rn} > \zeta_{rn}) \).
This section shows how the theory of the previous sections is implemented in MATLAB/simulink. The full model can be seen in figure 7.1. In the following sections each of the colored blocks is described, starting with the initialization of the simulink model.

7.1. Initialization

The simulink model is started using a matlab script in which all parameters for the simulation are set. The reference spacecraft orbit is defined in mean orbital elements and is then converted to osculating orbit elements. The relative spacecraft orbit is defined in mean ROE and is then converted to classical orbit elements and finally to osculating orbit elements. The desired reference is defined in mean ROEs.

```matlab
1 2 3 4 5
% % Init Constrained Formation Flying Model
% %
% % Credit: This code is generated using function of S3 programmed
% % by Duncan Eddy, Stanford University
6
7
8 clear all;
9 clc
10 load('eop_data_final')
11 load('ggm01s')
12
13 mass = 1.3; % kg
14
15 d2r = pi/180; % degree to radian conversion
16 % center spacecraft mean orbit definition
17 a = 6892e3; % [m]
18 e = 0.0001384;
19 i = 97.4*d2r;
20 OMEGA = 266.1539*d2r;
21 omega = 89.1198*d2r;
22 M = 45.88*d2r; % initial mean anomaly
23
24 oec_mean = [a e i OMEGA omega M];
25 oec_osc = sff_mean2osc(oec_mean); %convert mean to osc
26
27 % relative spacecraft mean ROE definition in [m]
28 offset = 0;
29 ada = 0+offset;
30 adlambda = 712;
31 adex = 400;
32 adey = 400;
33 adix = 100;
```
Figure 7.1: Simulink Model including full dynamic propagation, pseudo measurement, Lyapunov controller and a reference governor.

adiy = 500;
roe = [ada adlambda adex adey adix adiy]';
roe_mean = sff_roe2oe(oec_mean, roe); % convert ROE to mean oe
oed_mean = sff_roe2oe(oec_mean, roe); % convert mean to osc

% number of orbits to estimate drift of semi-major axis:
drift_estimation = 3;
% number of orbits for reconfiguration after estimation of semi-major axis
% drift
orbits = 35;
sim_step = 20;           % simulation time step size in [s]
sim_end = (orbits+3)*5650;   % end time of simulation [s]

% reference ROE state definition in [m]
ada = 0+ offset;
adlambda = 212;
adex = 200;
adey = 200;
adix = 0;
adiy = 200;

roe_ref = [ada adlambda adex adey adix adiy]';

% define feedback matrix parameters
N = 14;
nom = 500;
position_variance = 10;
velocity_variance = .1;

% set extended kalman filter
P = 1*eye(6);
7.2. Propagator

The propagator can be seen in figure 7.2. The main blocks satellite 1 and satellite 2 are provided by the S3 toolbox. Included force models are perturbation effects caused by third body moon, third body Sun, solar radiation pressure, aerodynamic drag, relativity, tides, empirical accelerations. The Earth's gravity model is applied using Degree and Order of 120. The control inputs generated by the controller of the relative spacecraft can be included as empirical accelerations provided in the RTN frame.

7.2.1. Pseudo Measurement and State Estimation

The pseudo measurement block (figure 7.3) has four inputs; two spacecraft positions in the ECI reference frame and three noise sources. Using S3 the osculating ECI positions are first converted to osculating classical orbit elements and then into mean orbit elements. The mean orbit elements are used to determine the ROE state. Gaussian Noise with a variance of 10 m is added to the true ROE state and an extended Kalman filter (figure 7.4) is used to perform relative state estimation. Absolute state estimation is under development at the University of Würzburg and will be implemented for a proper simulation of the NetSat mission. In this case the noise source will be added as noise in the Earth-centered Earth-fixed (ECEF) reference frame.

```matlab
function [mean_roe_updated, P_updated, new_time, z, y] = fcn(mean_oe_chief, u_RTN_dep, u_RTN_chief, ProcessNoiseCovVec, MeasNoiseCovVec, Measurement, x_apriori, P_apriori, t, old_time)
coder.extrinsic('s3_meantoecc');
coder.extrinsic('s3_ecctotrue');
```
Figure 7.3: Pseudo measurement of the relative state.

Figure 7.4: Relative State Estimation using an extended Kalman filter.
mean_roe_updated = zeros(6,1);  
z = zeros(6,1);  
y = zeros(6,1);  
P_updated = zeros(6,6);  
%simstep = 0;  
J2 = 1.08263e-3;  
R_EARTH = 0.6378136300e7;  
MU_EARTH = 0.3986004415e15;  
new_time = t;  
simstep = t - old_time;  
a = mean_oe_chief(1);  
e = mean_oe_chief(2);  
inc = mean_oe_chief(3);  
OMEGA = mean_oe_chief(4);  
omega = mean_oe_chief(5);  
M = mean_oe_chief(6);  
PHI = eye(6);  
Q2 = diag(ProcessNoiseCovVec);  
R = diag(MeasNoiseCovVec);  
if t==0  
   x_apriori = Measurement;  
end  
% state transition matrix PHI  
n = sqrt(MU_EARTH/a^3);  
eta = sqrt(1-eta^2);  
gamma = 3/4+J2*(R_EARTH)^2*sqrt(MU_EARTH);  
E = 1+eta;  
F = 4+3-eta;  
G = 1/eta^2;  
P2 = 3*cos(eta)^2-1;  
Q = 5*cos(eta)^2-1;  
S = sin(2*eta);  
T = sin(eta)^2;  
kappa = gamma/a^4*(7/2)/eta^4;  
dot_omega = kappa*Q;  
omega_f = omega + dot_omega*simstep;  
ey0 = e*sin(omega);  
ex0 = e*cos(omega);  
eyf = e*sin(omega_f);  
exf = e*cos(omega_f);  
PHI(2,1) = -7/2*kappa*E*P2*simstep-3/2*n*simstep;  
PHI(3,1) = 7/2*kappa*eyf*Q*T;  
PHI(4,1) = -7/2*kappa*exf*Q*T;  
PHI(5,1) = 7/2*kappa*S*simstep;  
PHI(2,3) = kappa*ex0*F*G*P2*simstep;  
PHI(3,3) = cos(dot_omega*simstep) -4*kappa*ex0*eyf*G*Q*simstep;  
PHI(4,3) = sin(dot_omega*simstep) +4*kappa*ex0*exf*G*Q*simstep;  
PHI(6,3) = -4*kappa*ex0*G*S*simstep;  
PHI(2,4) = kappa*ey0*F*G*P2*simstep;  
PHI(3,4) = -sin(dot_omega*simstep) -4*kappa*ey0*eyf*G*Q*simstep;  
PHI(4,4) = cos(dot_omega*simstep) +4*kappa*ey0*exf*G*Q*simstep;  
PHI(6,4) = -4*kappa*ey0*G*S*simstep;  
PHI(2,5) = -kappa*F*S*simstep;  
PHI(3,5) = 5*kappa*eyf*S*simstep;  
PHI(4,5) = -5*kappa*exf*S*simstep;  
PHI(6,5) = 2*kappa*T*simstep;  
% control input matrix  
E = s3_meanomega(Me);  
f = 0;  
f = s3_eccotottrue(E,e);  
theta = f*omega;  
ex = (ex0+exf)/2;  
ey = (ey0+eyf)/2;
7.3. Feedback Control

The Lyapunov feedback control of the reduced model can be seen in figure 7.5. The blue box shows the calculation of the reduced state error. The yellow box has multiple sub-
functions to calculate the plant matrix $A$ (equation (A.2)), the control input matrix $B$ (equation (A.8)) and the feedback gain matrix $P$ (equation (4.4)). The controller block simply calculates the desired control inputs based on the control law given in equation (4.1) and its C++ S-function code is given below.

```cpp
// filename.cpp

#include <iostream>
#include "s3_vector.h"
#include "s3_matrix.h"
#include <cmath>
#include "simstruc.h"

#define PI 3.14159265358979323846

static void mdlInitializeSizes(SimStruct *S) {
    ssSetNumSFcnParams(S, 0);
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
        return; /* Parameter mismatch reported by the Simulink engine*/
    }
    if (!ssSetNumInputPorts(S, 7)) return;
    ssSetInputPortDirectFeedThrough(S, 0, 1);
    ssSetInputPortDirectFeedThrough(S, 1, 1);
    ssSetInputPortDirectFeedThrough(S, 2, 1);
    ssSetInputPortDirectFeedThrough(S, 3, 1);
    ssSetInputPortDirectFeedThrough(S, 4, 1);
    ssSetInputPortDirectFeedThrough(S, 5, 1);
    ssSetInputPortDirectFeedThrough(S, 6, 1);
    if (!ssSetInputPortMatrixDimensions(S, 0, 5, 5)) {
        return; }
    ssSetInputPortVectorDimension(S, 1, 5);
    ssSetInputPortMatrixDimensions(S, 2, 2, 5);
    ssSetInputPortVectorDimension(S, 3, 5);
    ssSetInputPortVectorDimension(S, 4, 6);
    ssSetInputPortMatrixDimensions(S, 5, 5, 5);
    ssSetInputPortVectorDimension(S, 6, 5);
    if (!ssSetNumOutputPorts(S, 1)) return;
    // if (!ssSetOutputPortMatrixDimensions(S, 0, 5, 5)) {
    //     return; }
    if (!ssSetOutputPortMatrixDimensions(S, 0, 5, 5)) {
        return; }
    ssSetOutputPortMatrixDimensions(S, 1, 3);
    // ssSetOutputPortMatrixDimensions(S, 2, 1);
    // ssSetOutputPortMatrixDimension(S, 3, 1);
    ssSetNumSampleTimes(S, 1);
    // Take care when specifying exception free code – see sfuntmpl.doc */
    // ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE);
}

static void mdlInitializeSampleTimes(SimStruct *S) {
    ssSetSampleTime(S, 0, INHERITED_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);
}
```
static void mdlOutputs(SimStruct *S, int_T tid)
{
    InputRealPtrsType Amatrix = ssGetInputPortRealSignalPtrs(S,0);
    InputRealPtrsType ROE = ssGetInputPortRealSignalPtrs(S,1);
    InputRealPtrsType BINV= ssGetInputPortRealSignalPtrs(S,2);
    InputRealPtrsType ERROR= ssGetInputPortRealSignalPtrs(S,3);
    InputRealPtrsType OE_CHIEF= ssGetInputPortRealSignalPtrs(S,4);
    InputRealPtrsType PFEED= ssGetInputPortRealSignalPtrs(S,5);
    //InputRealPtrsType xRTN = ssGetInputPortRealSignalPtrs(S,1);
    double* out_dep = (double*) ssGetOutputPortRealSignal(S,0);
    //double* out_chief = (double*) ssGetOutputPortRealSignal(S,1);
    //double* out2 = (double*) ssGetOutputPortRealSignal(S,2);
    //double* out3 = (double*) ssGetOutputPortRealSignal(S,3);

    Vector roe(5), error(5), oe_chief(6), drag(5), u_2(2);//, u_2(2);
    Matrix Aplant(5,5), Binv(2,5), P(5,5);
    double e_pass, M_pass, w_pass, tol, E0, E1, f;

    for (int i=0; i<6; i++)
    {
        oe_chief(i) = *OE_CHIEF[i];
    }

    for (int i=0; i<5; i++)
    {
        drag[i] = *myDrag[i]/oe_chief(0);
        roe[i] = *ROE[i]/oe_chief(0);
        error[i] = *ERROR[i]/oe_chief(0);
        for (int j=0; j<5; j++)
        { // Aplant = Amatrix +5*OE CHIEF[];
            P(i,j) = *PFEED[i+5+j];
            if (i<2)
                Binv(i,j) = *BINV[i+2+j];
        }
    }

    u_2 = -Binv*(Aplant*roe+drag+P*error);

    //Vector control_out(3);
    //control_out(0) = error(1);
    //control_out(1) = error(2);
    //control_out(2) = 0;
    //Vector test(3);
    out_dep[0] = 0;
    out_dep[1] = u_2(0);
    out_dep[2] = u_2(1);

    //out_dep[0] = 0;
    //out_dep[1] = u_2(0);
    //out_dep[2] = u_2(1);
    //out_chief[0] = 0;
    //out_chief[1] = -u_2(0)/2;
    //out_chief[2] = -u_2(1)/2;
    //out2[0] = sqrt(pow(out dep[0], 2)+pow(out dep[1], 2)+pow(out dep[2], 2));
    //out3[0] = out2[0]/1.3;
    /*
    for (int i=0; i<5; i++)
    { // for (int j=0; j<5; j++)
        out3[ColMajor(i,j,5,5)] = Aplant[i,j];
    }
*/
7.3. Feedback Control

Figure 7.6: Reference Governor Simulink block.

7.3.1. Reference Governor
The reference governor (RG) can be seen in figure 7.6. It is programmed to enforce and satisfy the discussed constraints. The RG is initialized with the initial reduced mean ROE state. The RG calculates the rate and direction of change of the applied reference which is then integrated. One can see at the top right of figure 7.6 how the integrated applied reference is then forwarded to the output of this block. It is shown that the RG in fact only affects four of the five states of the reduced model, which correspond to the eccentricity and the inclination vectors. The applied reference for the difference in the semi-major axis is calculated in the same RG Simulink block, but actually is the solution of equation (4.15) in order to control the along-track separation \( \delta \lambda \).

```
function [da_des, da, reset, orbit_counted_new, counter_new, lambda_start, da_variation, u_ip, in_plane, switched_new] = fcn(roe_mean, ref_des, oe, ref_app, N, nom, orbit_counted, orbit_count, t, counter_old, lambda_save, orbits, drift_est, m, switched_old)

switched_new = 0;

% initialize variables
coder.extrinsic('s3_meantoecc ');
encoder.extrinsic('s3_eccottrue ');
da = zeros(5.1); % derivative of applied reference
counter_new = 0;
dv_ip = 0;
u_ip = 0;
in_plane = 1;
```
\begin{verbatim}
da_variation = 0;
P = zeros(5, 5);
counter_new = counter_old + 1;
lambda_start = 0;
da_des = 0;
orb_counts = new = 1;
reset = 0;

% required constants
GM_EARTH = 0.3986004415e15;
% mean motion
n = sqrt(GM_EARTH/oe(1)^3);

% reduced desired reference
ref_des_red = [ref_des(1);ref_des(3);ref_des(4);ref_des(5);ref_des(6)];
% reduced actual ROE mean state
roe_red = [roe_mean(1);roe_mean(3);roe_mean(4);roe_mean(5);roe_mean(6)];
% absolute error between desired reference and currently applied reference
ref_err = ref_app-ref_des_red;
% actual error
err_actual = roe_red-ref_des_red;
% GLOBAL POTENTIAL
% if the difference between the applied and the desired reference is
% greater than eta=1 m the global potential direction is unified.
% if it is smaller than 1 m it tends to zero
eta = 1; % see explanation above

% equation 46
if norm(ref_app-ref_des_red) >= eta
    \eta = 1-
\end{verbatim}
% collision safety in rt–plane is not provided. Now, check whether
% the actual along–track separation is negative or positive such
% that the proper safety distance can be set for the temporary new
% reference separation, equation 73
if \( roe_{mean} \%\) < 0
\[
\text{ref\_des\%} = \min ([ -2 \times \text{norm} \% \text{de} \% - \epsilon \% \lambda_{save} \% \text{ref\_des\%} ) ] ;
\]
else
\[
\text{ref\_des\%} = \max ([ 2 \times \text{norm} \% \text{de} \% + \epsilon \% \lambda_{save} \% \text{ref\_des\%} ) ] ;
\]
end
% safe this distance to know it during the next time step
\text{lambda\_start} = \text{ref\_des\%} ;
% calculate the magnitude of the potential field (the directional
% information is given into the negative direction of
% the attractive the global potential field), equation 71
\text{magnitude\_1} = -U_{\text{psi}} \% \text{rn} \% ^{\%2}+ \% (ceta \% \text{rn} \% - \% c \% \text{rn} \% ) \% ^{\%2}/(ceta \% \text{rn} \% ^{\%2} - U_{\text{psi}} \% \text{rn} \% ^{\%2}) / \% \text{c} \% \text{rn} \% ;
% this is just a safety check. By definition this magnitude can
% never become positive as long as equation 71 is only applied for
% inside the influence region. But its a numerical simulation, just
% to be sure, since a positive value would have severe impacts on
% guidance. Preclusio:
if \( \text{magnitude\_1} \% > \% 0 \)
\text{magnitude\_1} = 0 ;
end
% calculate the actual error of the actual along–track separation to
% the newly defined safety separation
\text{c\_lam} = \text{roe\_mean\%} - \text{ref\_des\%} ;
% if the error is smaller than 25 m proceed with de, di
% reconfiguration, otherwise stop it. The 25 m are defined because
% the along–track separation is only passively controlled by
% leveraging Keplerian dynamics. The accuracy to control that
% along–track separation is about \% 8\%m
if \( \text{abs} \% \text{c\_lam} \% > \% 25 \)
\text{pot\_rn} = \max ([ \text{magnitude\_1} \% ; \% -1 \% ) \% \times \% \text{pot} \% ;
else
\text{pot\_rn} = \text{zeros} \% (5,1) ;
end
else
\text{pot\_rn} = \text{zeros} \% (5,1) ;
if \( \text{roe\_mean\%} \% < \% 0 \)
\text{ref\_des\%} = \min ([ -2 \times \text{norm} \% \text{de} \% - \epsilon \% \lambda_{save} \% \text{ref\_des\%} ) ] ;
else
\text{ref\_des\%} = \max ([ 2 \times \text{norm} \% \text{de} \% + \epsilon \% \lambda_{save} \% \text{ref\_des\%} ) ] ;
end
\text{lambda\_start} = \text{ref\_des\%} ;
end
% if this constraint is not desired for a simulation, uncomment the next
% line:
% \% pot\_rn = \% inf ;
end
% EXAMPLE CONSTRAINT 1:
% circular no-entry–zone constraint for the eccentricity vector
U_{\text{psi}} = 15 ; \% all before the actual constrained area the potential
\% field increases significantly
\text{zeta} = 100 ; \% all before the actual constrained area the potential
\% will be added to the global potential
\text{r\_1} = 40 ; \% 40 m radius of no-entry–zone constraint
\text{x\_1} = [0;350] ; \% center of constrained area
\text{G\_1} = [0 0 0 1 0 ; 0 0 0 0 1] ; \% equation 6, defining that the
\% inclination is enforced
\text{C\_1} = \text{norm} \% \{(G\_1\% \times \text{ref}\_app\% - \% x\_1\% ) \% - \% r\_1\% \} ; \% equation 41, constraint form 6
if \( \text{C\_1} \% < \% \text{zeta} \)
\text{nu\_1} = \text{G\_1\% \times \text{ref}\_app\% - \% x\_1\% } / \% \text{norm} \% \{(G\_1\% \times \text{ref}\_app\% - \% x\_1\% ) \% ; \%}
% equation 43
\texttt{pot}_1 = -\text{Upsilon}^2 \ast (\text{zeta}^2 - \text{C}_1^2) /(-\text{Upsilon}^2 + \text{zeta}^2) / \text{C}_1^2 \ast \text{nu}_1;
\textbf{else}
\text{pot}_1 = \text{zeros}(5,1); \quad \% \text{equation 43}
\textbf{end}
\text{Gamma}_1 = 0.5 \ast (\text{norm}(\text{G}_1 \ast \text{ref}_\text{app} \ast \text{x}_1) - \text{r}_1) \ast \text{zeta}^2; \quad \% \text{equation 24}
\% \text{if you don't want to use this constraint, uncomment the next two lines}
\text{pot}_1 = \text{zeros}(5,1);
\text{Gamma}_1 = \text{inf};
\% \text{EXAMPLE CONSTRAINT 2:}
\% \text{wall constraint: dex} \gg 300 \text{ m}
\text{Upsilon} = 6; \quad \% \text{dex should stop at 310 m}
\text{zeta} = 10; \quad \% \text{dex applied reference starts being}
\% \text{affected at dex} = 300 \ast 50 = 350 \text{ m}
\text{c}_2 = [0 1 0 0 0]'; \quad \% \text{equation 9}
\text{d}_2ip = 294; \quad \% \text{equation 9}
\text{C}_2 = -\text{d}_2ip \ast \text{c}_2 \ast \text{ref}_\text{app}; \quad \% \text{equation 41, constraint form 9}
\text{nu}_2 = \text{c}_2; \quad \% \text{equation 48, constraint form 9}
\textbf{if} \text{c}_2 < \text{zeta}
\text{pot}_2 = -\text{Upsilon}^2 \ast (\text{zeta}^2 - \text{C}_2^2) /(-\text{Upsilon}^2 + \text{zeta}^2) / \text{C}_2^2 \ast \text{nu}_2;
\textbf{else}
\text{pot}_2 = \text{zeros}(5,1); \quad \% \text{equation 43}
\textbf{end}
\% \text{threshold for constraint i=2}
\text{Gamma}_2 = 0.5 \ast (\text{c}_2^2 \ast \text{ref}_\text{app} - \text{d}_2ip) \ast \text{zeta}_2; \quad \% \text{equation 22}
\% \text{if you don't want to use this constraint, uncomment the next two lines}
\text{pot}_2 = \text{zeros}(5,1);
\% \text{Gamma}_2 = \text{inf};
\% \text{Maximum Thrust Constraint:}
\% \text{maximum thrust is here set to 12e-0N}
\text{err} = \text{roe}_\text{red} \ast \text{ref}_\text{app};
\text{error} = \text{err};
\text{dv}_\text{ip} = \text{n} \ast 0.5 \ast \text{sqrt} ((\text{err}_\text{actual}(2)) \ast 2 + (\text{err}_\text{actual}(3)) \ast 2);
\text{e} = \text{oe}(2); \quad \% \text{equation 4}
\text{orbits}_\text{ip} = \text{norm}(\text{err}_\text{actual}(2:3)) / (2 \ast (1 - e) \ast \text{norm}(\text{err}_\text{actual}(4:5)) + \text{norm}(\text{err}_\text{actual}(4:5))) \ast \text{orbits};
\text{orbits}_\text{oop} = 2 \ast (1 - e) \ast \text{norm}(\text{err}_\text{actual}(4:5)) / (2 \ast (1 - e) \ast \text{norm}(\text{err}_\text{actual}(4:5)) + \text{norm}(\text{err}_\text{actual}(2:3))) \ast \text{orbits};
\% \text{time left for reconfiguration}
\text{orbits}_\text{left}_\text{ip} = \text{orbits}_\text{ip} + \text{orbits}_\text{oop} \ast \text{drift}_\text{est} - \text{orbit}_\text{count};
\text{orbits}_\text{left}_\text{oop} = \text{orbits}_\text{oop} \ast \text{drift}_\text{est} - \text{orbit}_\text{count};
\% \text{if orbits}_\text{left}_\text{oop} < 1
\text{orbits}_\text{left}_\text{oop} = 1;
\textbf{end}
\% \text{if orbits}_\text{left}_\text{ip} < 1
\text{orbits}_\text{left}_\text{ip} = 1;
\textbf{end}
\text{Int}_1 = 1;
\text{for} \text{i=1:1:N/2-1}
\text{Int}_1 = \text{Int}_1 + ((\text{N} - (2 \ast (\text{i} - 1) + 1)) / (\text{N} - 2 \ast (\text{i} - 1)));
\textbf{end}
\text{Int} = \text{Int}_1 \ast \text{pi};
\text{T} = 2 \ast \text{pi} \ast \text{n};
\text{u}_\text{ip} = \text{dv}_\text{ip} \ast 2 \ast \text{pi} / \text{T} / \text{Int} / \text{orbits}_\text{left}_\text{ip};
\text{F}_\text{ip} = \text{u}_\text{ip} \ast \text{m};
\% \text{da variation}
\text{da}_\text{variation} = \text{Int}_1 \ast \text{u}_\text{ip} \ast \text{oe}(1) / 2;
\% \text{da variation}
\text{da}_\text{variation} = \text{dv}_\text{ip} / \text{orbits}_\text{left}_\text{ip} \ast \text{n};
\% \text{da max}
\text{da}_\text{max} = \text{da}_\text{variation} / 2;
\text{dv}_\text{oop} = \text{n} \ast \text{sqrt} ((\text{err}_\text{actual}(4)) \ast 2 + (\text{err}_\text{actual}(5)) \ast 2);
\text{u}_\text{oop} = \text{dv}_\text{oop} \ast 2 \ast \text{pi} / \text{T} / \text{Int} / \text{orbits}_\text{left}_\text{oop};
7.3. Feedback Control

F_oop = u_oop\cdot m;

\% thrust constraint
\%
\% u_ip = 20e-6;
\% u_oop = 20e-6;
\% orbits_left_oop = 0;
da_variation = u_ip\cdot T/2/n\cdot Int1;
da_max = da_variation/2;
\%
\%
da_variation2 = dv_ip/n;
da_max2 = da_variation2/2;

if da_variation2<da_variation
  da_variation = da_variation2;
da_max = da_max2;
end
\%
%
\% optimal mean argument of latitudes to apply thrust
u_ip_opt = atan2(-error(3),-error(2))\cdot pi;
u_oop_opt = atan2(-error(5),-error(4));

a_pass = oe(1);
e_pass = oe(2);
i_pass = oe(3);
omega_pass = oe(5);
M_opt = u_ip_opt-omega_pass;
E_act_ip = s3_meantoecc(M_opt,e_pass);
E_act_oop = s3_meantoecc(-omega_pass+u_oop_opt,e_pass);
f = 0;
f = s3_eccotrue(E_act_ip,e_pass);
eta_act = sqrt((1-e_pass^2));
theta_ip = f\cdot omega_pass;
e_x = e_pass\cdot cos(omega_pass);
e_y = e_pass\cdot sin(omega_pass);

\% control input matrix
B = zeros(5,2);
B(1,1) = 2/eta_act*(1+e_pass\cdot cos(f));
B(2,1) = eta_act*[(2+e_pass\cdot cos(f))\cdot cos(theta_ip) + e_x]/(1+e_pass\cdot cos(f));
B(3,1) = eta_act*[(2+e_pass\cdot cos(f))\cdot sin(theta_ip) + e_y]/(1+e_pass\cdot cos(f));
B(2,2) = eta_act\cdot e_y\cdot sin(theta_ip)/tan(i_pass)/(1+e_pass\cdot cos(f));
B(3,2) = -eta_act\cdot e_x\cdot sin(theta_ip)/tan(i_pass)/(1+e_pass\cdot cos(f));
B(4,2) = eta_act\cdot cos(theta_ip)/tan(i_pass)/(1+e_pass\cdot cos(f));
B(5,2) = eta_act\cdot sin(theta_ip)/(1+e_pass\cdot cos(f));

B = 1/a_pass/n\cdot B;
Binv_ip = pinv(B);

\% equation 26
d_2ip = u_ip\cdot norm(Binv_ip*A\cdot ref_app/oe(1));
f = s3_eccotrue(E_act_oop,e_pass);
theta = f\cdot omega_pass;
B = zeros(5,2);
B(1,1) = 2/eta_act*(1+e_pass\cdot cos(f));
B(2,1) = eta_act*[(2+e_pass\cdot cos(f))\cdot cos(theta) + e_x]/(1+e_pass\cdot cos(f));
B(3,1) = eta_act*[(2+e_pass\cdot cos(f))\cdot sin(theta) + e_y]/(1+e_pass\cdot cos(f));
B(2,2) = eta_act\cdot e_y\cdot sin(theta)/tan(i_pass)/(1+e_pass\cdot cos(f));
B(3,2) = -eta_act\cdot e_x\cdot sin(theta)/tan(i_pass)/(1+e_pass\cdot cos(f));
B(4,2) = eta_act\cdot cos(theta)/tan(i_pass)/(1+e_pass\cdot cos(f));
B(5,2) = eta_act\cdot sin(theta)/(1+e_pass\cdot cos(f));

B = 1/a_pass/n\cdot B;
Binv_oop = pinv(B);
d_2oop = u_oop\cdot norm(Binv_oop*A\cdot ref_app/oe(1));
P=zeros(5,5);
if error(1)>0
  if cos(theta_ip-(u_ip+pi))>0
    ...
  end
  ...
else
  ...
end
\[
P(1,1) = \cos(\theta_{ip}-u_{ip})/N;\\
\text{else if } \cos(\theta_{ip}-u_{ip})>0\\
P(1,1) = \cos(\theta_{ip}-u_{ip})/N;\\
\text{else end}
\]

\[
P(1,1)=0;\\
\text{else if } \cos(\theta_{ip}-u_{ip})>0\\
P(1,1) = \cos(\theta_{ip}-u_{ip})/N;\\
\text{else end}
\]

\[
P(1,1)=0;\\
\text{else if abs(error(1))}>20\\
\text{else end}
\]

\[
P(1,1)=0;\\
\text{else if cos(theta_{ip}-u_{ip})>0}\\
P(1,2)=\cos(\theta_{ip}-u_{ip})/N;\\
P(1,3)=\cos(\theta_{ip}-u_{ip})/N;\\
\text{else end}
\]

\[
P(1,2)=0;\\
P(1,3)=0;\\
\text{else end}
\]

\[
P(1,2)=\cos(\theta_{ip}-u_{ip})/N;\\
P(1,3)=\cos(\theta_{ip}-u_{ip})/N;\\
\text{else end}
\]

\[
P(2,2)=\cos(\theta_{ip}-u_{ip})/N;\\
P(3,3)=\cos(\theta_{ip}-u_{ip})/N;\\
\text{else end}
\]

\[
P(2,2)=0;\\
P(3,3)=0;\\
\text{else end}
\]

\[
P(4,4)=\cos(\theta_{ip}-u_{oop})/N;\\
P(5,5)=\cos(\theta_{ip}-u_{oop})/N;\\
\text{else end}
\]

\[
\%P = \text{eye}(5);\\
P=P\times N;\\
\text{else the following section divides the constraint for the maximum thrust into}\\
\text{the in-plane and the out-of-plane. This is done because it is assumed}\\
\text{that thrusters are available in both the along-track direction and the}\\
\text{normal direction of the RTN frame. Therefore also equation 31 has to be}\\
\text{divided into those two regimes. I perform this by still applying equation}\\
\text{31 as given, but setting the the elements of the feedback gain matrix to}\\
\text{zero that do not belong to the in-plane motion if the threshold for the}\\
\text{thruster into the along-track direction is calculated. The same is done}\\
\text{with the error (also for equation 31)}
\]

\[
\text{P out}=P;\\
\text{error in}=\text{err};\\
\text{err in}(1)=0;\\
\text{err in}(4:5)=0;\\
\text{err out}=\text{err};\\
\text{err out}(1:3)=0;\\
\text{P out}=P;\\
\text{P out}(1:3,1:3)=0;\\
\text{P out}(4:5,4:5)=0;\\
\text{else end}
\]

\[
\% \text{equation 31}
\]

\[
\Gamma_{3oop} = 0.5 * (d_{2o} / \text{norm}(\text{Binv_oop} * \text{Pin} * \text{error in} / \text{norm} (\text{err in}) / \text{oe}(1))) ^ 2;\\
\Gamma_{3oop} = 0.5 * (d_{2o} / \text{norm}(\text{Binv_oop} * \text{Pout} * \text{error out} / \text{norm} (\text{err out}) / \text{oe}(1))) ^ 2;\\
\text{ui opt} = \text{atan}2(-\text{error}(3),-\text{error}(2));\\
\text{M opt} = \text{ui opt} - \text{omega pass};\\
\text{E_act_ip} = \text{s3_meantoecc(M opt, e_pass)};\\
f = \text{s3_eccottrue(E_act_ip, e_pass)};\\
\theta_{ip} = f + \omega \text{pass};\\
B = \text{zeros}(5,2);
\]
\[ B(1,1) = \frac{2}{\eta_{act}} \cdot (1 + e_{pass} \cdot \cos(f)) \]

\[ B(2,1) = \frac{\eta_{act} \cdot (1 + e_{pass} \cdot \cos(f)) \cdot \cos(\theta_{ip})}{\eta_{ip}} + e_{x} \cdot (1 + e_{pass} \cdot \cos(f)) \]

\[ B(3,1) = \frac{\eta_{act} \cdot (1 + e_{pass} \cdot \cos(f)) \cdot \sin(\theta_{ip})}{\eta_{ip}} + e_{y} \cdot (1 + e_{pass} \cdot \cos(f)) \]

\[ B(2,2) = \frac{\eta_{act} \cdot e_{y} \cdot \sin(\theta_{ip})}{\eta_{ip}} \cdot \tan(\eta_{pass}) \cdot (1 + e_{pass} \cdot \cos(f)) \]

\[ B(3,2) = -\frac{\eta_{act} \cdot e_{x} \cdot \sin(\theta_{ip})}{\eta_{ip}} \cdot \tan(\eta_{pass}) \cdot (1 + e_{pass} \cdot \cos(f)) \]

\[ B(4,2) = \frac{\eta_{act} \cdot \cos(\theta_{ip})}{\eta_{ip}} \cdot (1 + e_{pass} \cdot \cos(f)) \]

\[ B(5,2) = \frac{\eta_{act} \cdot \sin(\theta_{ip})}{\eta_{ip}} \cdot (1 + e_{pass} \cdot \cos(f)) \]

\[ \text{if} \quad \text{error}(1) > 0 \]
\[ \quad \text{if} \quad \text{cos}(\theta_{ip} - (u_{ip} + \pi)) > 0 \]
\[ \quad \quad \text{P}(1,1) = \text{cos}(\theta_{ip} - u_{ip}) \cdot \gamma; \]
\[ \quad \text{else} \]
\[ \quad \quad \text{P}(1,1) = 0; \]
\[ \text{end} \]

\[ \text{else} \]
\[ \quad \text{if} \quad \text{cos}(\theta_{ip} - u_{ip}) > 0 \]
\[ \quad \quad \text{P}(1,1) = \text{cos}(\theta_{ip} - u_{ip}) \cdot \gamma; \]
\[ \quad \text{else} \]
\[ \quad \quad \text{P}(1,1) = 0; \]
\[ \text{end} \]

\[ \text{end} \]

\[ \text{if} \quad \text{abs}(\text{error}(1)) > 20 \]
\[ \text{if} \quad \text{error}(1) < 0 \]
\[ \quad \text{if} \quad \text{cos}(\theta_{ip} - (u_{ip} + \pi)) > 0 \]
\[ \quad \quad \text{P}(2,2) = \text{cos}(\theta_{ip} - u_{ip}) \cdot \gamma; \]
\[ \quad \text{else} \]
\[ \quad \quad \text{P}(2,2) = 0; \]
\[ \quad \text{P}(3,3) = 0; \]
\[ \text{end} \]

\[ \text{else} \]
\[ \quad \text{if} \quad \text{cos}(\theta_{ip} - u_{ip}) > 0 \]
\[ \quad \quad \text{P}(2,2) = \text{cos}(\theta_{ip} - u_{ip}) \cdot \gamma; \]
\[ \quad \text{else} \]
\[ \quad \quad \text{P}(2,2) = 0; \]
\[ \quad \text{P}(3,3) = 0; \]
\[ \text{end} \]

\[ \text{end} \]

\[ \text{end} \]

\[ \text{P}(4,4) = \text{cos}(\theta_{ip} - \eta_{oop}) \cdot \gamma; \]

\[ \text{P}(5,5) = \text{cos}(\theta_{ip} - \eta_{oop}) \cdot \gamma; \]

\[ \text{P} = \text{P} \cdot \text{P} \cdot \text{P}; \]

\[ \text{Pin} = \text{P}; \]

\[ \text{err_in} = \text{err}; \]

\[ \text{err_in}(1) = 0; \]

\[ \text{err_in}(4:5) = 0; \]

\[ \text{err_out} = \text{err}; \]

\[ \text{err_out}(1:3) = 0; \]

\[ \text{Pout} = \text{P}; \]

\[ \text{Pout}(1:3,1:3) = 0; \]

\[ \text{Pin}(4:5,4:5) = 0; \]

\[ \text{Gamma}_3\text{ip}_{\text{new}} = 0.5 \cdot \text{norm}((\text{Binv_ip} \cdot \text{Pin}) \cdot \text{err_in} \cdot \text{norm}((\text{err_in}) \cdot \text{oe}(1))) \cdot 2; \]

\[ \text{if} \quad \text{Gamma}_3\text{ip}_{\text{new}} = \text{Gamma}_3\text{ip} \]

\[ \quad \% \text{Gamma}_3\text{ip}_{\text{new}} = \text{Gamma}_3\text{ip}; \]

\[ \text{end} \]

\[ \text{end} \]

\[ \text{for} \quad f = 0:2/\pi; 20:2/\pi \]

\[ \quad \text{theta} = f \cdot \text{omega_pass}; \]

\[ \quad e_{x} = e_{pass} \cdot \text{cos}(\text{omega_pass}); \]

\[ \quad e_{y} = e_{pass} \cdot \text{sin}(\text{omega_pass}); \]

\[ \quad B = \frac{1}{2} \cdot \text{eta_act} \cdot (1 + e_{pass} \cdot \cos(f)) \cdot 0; \]

\[ \quad \text{eta_act} \cdot [(2 + e_{pass} \cdot \cos(f)) \cdot \cos(\theta_{ip}) + e_{x}] \cdot (1 + e_{pass} \cdot \cos(f)) \cdot \text{eta_act} \cdot e_{y} \cdot \sin(\theta_{ip}) \cdot \tan(\eta_{pass}) \cdot (1 + e_{pass} \cdot \cos(f)); \]
\[
\eta_{\text{act}} = \frac{(2 + e_{\text{pass}} \cdot \cos(f)) \cdot \sin(\theta) + e_y}{(1 + e_{\text{pass}} \cdot \cos(f))} - \eta_{\text{act}} \cdot e_x \cdot \sin(\theta) \cdot \tan(\iota_{\text{pass}})
\]

\[
\begin{align*}
0 &= \eta_{\text{act}} \cdot \cos(\theta) / (1 + e_{\text{pass}} \cdot \cos(f)) \cdot \tan(\iota_{\text{pass}}) \\
0 &= \eta_{\text{act}} \cdot \sin(\theta) / (1 + e_{\text{pass}} \cdot \cos(f)) \cdot \tan(\iota_{\text{pass}})
\end{align*}
\]

\[
B = 1 / a_{\text{pass}} / n \cdot B
\]

\[
\text{Binv} = \text{pinv}(B)
\]

\[
\text{if} \ \cos(\theta - (u_{\text{ip}} + \pi)) > 0
\]
\[
\begin{align*}
P(1,1) &= \cos(\theta - u_{\text{ip}}) / N \\
P(1,1) &= \cos(\theta - u_{\text{ip}}) / N
\end{align*}
\]

\[
\text{else}
\]
\[
\begin{align*}
P(1,1) &= 0 \\
P(1,1) &= 0
\end{align*}
\]

\[
\text{end}
\]

\[
\text{if} \ \cos(\theta - u_{\text{ip}}) > 0
\]
\[
\begin{align*}
P(2,2) &= \cos(\theta - u_{\text{ip}}) / N \\
P(2,2) &= \cos(\theta - u_{\text{ip}}) / N
\end{align*}
\]

\[
\text{else}
\]
\[
\begin{align*}
P(2,2) &= 0 \\
P(2,2) &= 0
\end{align*}
\]

\[
\text{end}
\]

\[
\text{if} \ \cos(\theta - u_{\text{ip}}) > 0
\]
\[
\begin{align*}
P(3,3) &= \cos(\theta - u_{\text{ip}}) / N \\
P(3,3) &= \cos(\theta - u_{\text{ip}}) / N
\end{align*}
\]

\[
\text{else}
\]
\[
\begin{align*}
P(3,3) &= 0 \\
P(3,3) &= 0
\end{align*}
\]

\[
\text{end}
\]

\[
\text{if} \ \text{abs}(\text{error}(1)) > 20
\]
\[
\text{if} \ \text{error}(1) > 0
\]
\[
\begin{align*}
P(2,2) &= \cos(\theta - u_{\text{ip}}) / N \\
P(2,2) &= \cos(\theta - u_{\text{ip}}) / N
\end{align*}
\]

\[
\text{else}
\]
\[
\begin{align*}
P(2,2) &= 0 \\
P(2,2) &= 0
\end{align*}
\]

\[
\text{end}
\]

\[
\text{else}
\]
\[
\begin{align*}
P(3,3) &= \cos(\theta - u_{\text{ip}}) / N \\
P(3,3) &= \cos(\theta - u_{\text{ip}}) / N
\end{align*}
\]

\[
\text{end}
\]

\[
\text{end}
\]

\[
\text{end}
\]

\[
\text{end}
\]

\[
\text{end}
\]

\[
\text{end}
\]

\[
\text{end}
\]

\[
\text{if} \ \text{Gamma}_{3ip} \new < \text{Gamma}_{3ip}
\]
\[
\text{Gamma}_{3ip} = \text{Gamma}_{3ip} \new
\]

\[
\text{end}
\]

\[
\text{end}
\]

\[
\text{end}
\]

\[
\% \ Gamma_{3ip} = \inf;
\]
\[
\% \ Gamma_{3oop} = \inf;
\]

\[
\% \ \text{CONTROL} \ \text{dot}(\text{delta lambda})
\]
\[
\% \ [\text{this part does not explicitly belong to the Reference Governor}]
\]
\[
\% \ \text{error in delta lambda}
\]
\[
\% \ DL = \text{me}(\text{mean}(2)) - \text{ref}_\text{des}(2)
\]
\[
\% \ \text{construct reduced actual and desired ROE state without delta lambda}
\]
\[
\text{kappa2} = 20000; \ \% \ \text{pseudo time}
\]
\[
\text{if} \ \text{norm} \ (\text{error}_{\text{actual}}) < 20
\]
7.3. Feedback Control

\[ \text{kappa} = 4000; \]
\end
\text{ddl\_max} = \text{da\_max} + 3/2 \times n; \quad \% \text{max\_max} \text{ allowed dot(\text{delta lambda}) equation 15}
\begin{align*}
\text{if} & \quad \text{Dl} > 0 \\
\text{ddl} = & \quad -\min(\text{abs} (\text{Dl}) / \text{kappa}2, \text{ddl\_max}); \quad \% \text{equation 17} \\
\text{else} & \\
\text{ddl} = & \quad \min(\text{abs} (\text{Dl}) / \text{kappa}2, \text{ddl\_max}); \quad \% \text{equation 17} \\
\text{end} \\
\text{da\_des} = & \quad -2 \times \text{ddl} / 3 \times n; \quad \% \text{equation 16 solved for delta a}
\end{align*}
\begin{align*}
\% \text{threshold 4} \\
\text{Upsilon} = & \quad 6; \quad \% \text{diy} \text{ should stop at 310 m} \\
\text{zeta} = & \quad 10; \quad \% \text{diy applied reference starts being} \\
\text{affected at dix} = 300 + 20 = 320 m
\end{align*}
\begin{align*}
\text{c}_4 = & \quad [0 0 0 0 1]'; \quad \% \text{equation 9} \\
\text{d}_4 = & \quad 394; \quad \% \text{equation 9} \\
\text{C}_4 = & \quad \text{d}_4 + \text{c}_4 \times \text{ref\_app}; \quad \% \text{equation 41, constraint form 9} \\
\text{nu}_4 = & \quad \text{c}_4; \quad \% \text{equation 48, constraint form 9} \\
\text{if} & \quad \text{C}_4 < \text{zeta} \\
\text{pot}_4 = & \quad \text{Upsilon}^2 \times (\text{zeta}^2 - \text{C}_4^2) / (\text{Upsilon}^2 + \text{zeta}^2) / \text{C}_4^2 + \text{nu}_4; \quad \% \text{equation 43} \\
\text{else} & \\
\text{pot}_4 = & \quad \text{zeros}(5, 1); \quad \% \text{equation 43} \\
\text{end}
\end{align*}
\begin{align*}
\% \text{vector field} \\
\text{rho} = & \quad \text{pot} - \text{pot\_2 - pot\_4}; \quad % \text{pot\_rn}; \quad % \text{pot\_2 - pot\_4} \quad \% \text{equation 51} \\
\text{error\_inplane} = & \quad \text{err}(1:3) \\
\text{error\_crossplane} = & \quad \text{err}(4:5) \\
\text{V\_inplane} = & \quad 0.5 \times (\text{error\_inplane}' \times \text{error\_inplane}); \\
\text{V\_crosstrack} = & \quad 0.5 \times (\text{error\_crossplane}' \times \text{error\_crossplane});
\end{align*}
\begin{align*}
\text{kappa} = & \quad .01 \times \text{eye}(5); \quad \% \text{arbitrary positive scalar} \\
\text{if} & \quad t = 0 \\
\text{Gamma}_3\text{ip} = & \quad \text{inf}; \\
\text{Gamma}_3\text{oop} = & \quad \text{inf}; \\
\text{end} \\
\% \text{in plane reference: Lyapunov threshold} \\
\% \text{check whether constraint i=2 is affecting the in–plane motion} \\
\text{if norm}([0 1 1 0 0] \times c_2) \\
\text{Gamma}_2\text{ip} = & \quad \text{Gamma}_2; \\
\text{else} & \\
\text{Gamma}_2\text{ip} = & \quad \text{inf}; \\
\text{end} \\
\% \text{check whether constraint i=1 is affecting the in–plane motion} \\
\%\% \\
\text{if norm}([0 1 1 0 0] \times G_1) \\
\text{Gamma}_1\text{lp} = & \quad \text{Gamma}_1; \\
\text{else} & \\
\text{Gamma}_1\text{lp} = & \quad \text{inf}; \\
\text{end} \\
\% \text{check whether constraint i=4 is affecting the in–plane motion} \\
\%\% \\
\text{if norm}([0 1 1 0 0] \times c_4) \\
\text{Gamma}_4\text{ip} = & \quad \text{Gamma}_4; \\
\text{else} & \\
\text{Gamma}_4\text{ip} = & \quad \text{inf}; \\
\text{end} \\
\text{Gamma} = & \quad \min([\text{Gamma}_2\text{ip}; \text{Gamma}_3\text{ip}; \text{Gamma}_4\text{ip}]); \quad \% \text{equation 51} \\
\text{paral} = & \quad \max(\text{Gamma\_V\_inplane}, 0); \quad \% \text{check how close the relative state}
if para1 == inf  % is to a constraint
    para1 = 1;
end
if norm(ref_err(2:3)) > 0.1
    da = kappa * para1 * rho; % equation 20
    da(4:5) = 0;
else
    da = zeros(5, 1);
end
switched = switched_old;
if (norm(err_actual(4:5)) > 0.1 || (orbit_count - orbits_left_oop) < 0) & & switched % 314 = 4 before
    da = zeros(5, 1);
in_plane = 0;
switched_new = 1;
end
if (norm([0 0 0 1 1] * c_2) * Gamma_2oop = Gamma_2;
else
    Gamma_2oop = inf;
end
% check whether constraint i=1 is affecting the out-of-plane motion
Gamma_3oop = Gamma_1;
if norm([0 0 0 1 1] * G_1)
    Gamma_1oop = Gamma_1;
else
    Gamma_1oop = inf;
end
% check whether constraint i=4 is affecting the in-plane motion
if norm([0 0 0 1 1] * c_4)
    Gamma_4oop = Gamma_4;
else
    Gamma_4oop = inf;
end
Gamma = min([Gamma_2oop; Gamma_3oop; Gamma_4oop]); % equation 51
Gamma_out = Gamma_4oop;
Gammaoop = Gamma;
if para1 == inf  % is to a constraint
    para1 = 1;
end
if norm(ref_err(4:5)) > 0.1
    da = kappa * para1 * rho; % equation 20
    da(1:3) = 0;
else
    da = zeros(5, 1);
end
da(1:3) = da_save;
% reset the reference governor once per orbit in order to take account of
% orbit perturbations. otherwise the realtive state would always have to
% follow a trajectory that is determined by the initial position. By
% resetting the reference governor this is avoided.
if ~reset
    if mod(orbit_count, 1) <= .1 & & ~orbit_counted
        reset = 1;
    end
    if mod(orbit_count, 1) > 1
        orbit_counted_new = 1;
    else
        orbit_counted_new = 0;
    end
end
if counter_new == 100
7.4. Thruster Block

The thruster block consists of two parts. The task distribution and the remaining fuel calculation (see figure 7.7). The task distribution makes sure that only thrusts are applied which are greater than 0.24\,\mu N, which corresponds to the smallest thrust level that can be performed by the nanoFEEP [3] thruster system. The remaining fuel calculation uses the nanoFEEP thruster characteristics to calculate how much fuel mass is required for a desired thrust level as a function of the corresponding specific impulse.
8.1. Thrust Constraint and Optimality

This example shows a reconfiguration from an initial state-space configuration to a desired one with only the thrust constraint being imposed on the relative state. The thrust is restricted to peak at $26 \mu N$. The initial configuration $a_c \delta \alpha_0$ and the desired configuration $a_c \delta \alpha_{\text{ref}}$ are given as

$$
\begin{align*}
    a_c \delta \alpha_0 &= \begin{pmatrix} a_c \delta a \\ a_c \delta \lambda \\ a_c \delta e_x \\ a_c \delta e_y \\ a_c \delta i_x \\ a_c \delta i_y \end{pmatrix}_0 = \begin{pmatrix} 0 \\ 712 \\ 400 \\ 400 \\ 20 \\ 500 \end{pmatrix} \text{[m]}, \\
    a_c \delta \alpha_{\text{ref}} &= \begin{pmatrix} a_c \delta a \\ a_c \delta \lambda \\ a_c \delta e_x \\ a_c \delta e_y \\ a_c \delta i_x \\ a_c \delta i_y \end{pmatrix}_{\text{ref}} = \begin{pmatrix} 0 \\ 212 \\ 200 \\ 200 \\ 0 \\ 200 \end{pmatrix} \text{[m]}.
\end{align*}
$$

The initial configuration is indicated in figure 8.6 as circles and the desired configuration is marked by black crosses. The initial chief’s classical orbit parameters are given as a near-circular low Earth regressive polar orbit as

$$
\alpha_{c,0} = \begin{pmatrix} a_c \\ e_c \\ i_c \\ \Omega_c \\ \omega_c \\ M_c \end{pmatrix}_0 = \begin{pmatrix} 6892 \text{[km]} \\ 0.0001384 \\ 97.400000^\circ \\ 266.153900^\circ \\ 89.119800^\circ \\ 45.880000^\circ \end{pmatrix}.
$$
The delta-\(v\) lower bound are derived with equations (5.7) and (5.8) from \(\Delta \delta e\) and \(\Delta \delta i\) as

\[
\Delta v_{lb} = \frac{a_{cn}}{\eta} \cdot \left( \frac{\|\Delta \delta e\|}{2} + (1 - e_c)\|\Delta \delta i\| \right) \\
= \frac{n}{\eta} \cdot \left( \frac{\|a_c \Delta \delta e\|}{2} + (1 - e_c)\|a_c \Delta \delta i\| \right) \\
= \frac{\sqrt{\mu/a_c^2}}{\sqrt{1 - e_c^2}} \left( \frac{1}{2} \sqrt{(a_c \Delta \delta e_x)^2 + (a_c \Delta \delta e_y)^2} + \sqrt{(a_c \Delta \delta i_x)^2 + (a_c \Delta \delta i_y)^2} \right) \\
= \frac{0.0011 \text{ [s}^{-1}] \left( \frac{1}{2} \sqrt{(200 \text{ [m]})^2 + (200 \text{ [m]})^2} + \sqrt{(20 \text{ [m]})^2 + (300 \text{ [m]})^2} \right)}{\sqrt{1 - 0.0001384^2}} \\
= 0.4878 \left[ \frac{\text{m}}{\text{s}} \right],
\]

with \(\mu = 3.986 \cdot 10^{14} \left[ \text{m}^3/\text{s}^2 \right]\). Figure 8.2 shows the actual fuel consumption of the reconfiguration, which is given by \(\Delta \nu = 0.5070 \text{ [m/s]}\). This fuel consumption corresponds to a 3.9\% higher fuel consumption than the optimal delta-\(v\) lower bound \(\Delta v_{lb}\). It is important to note that the lower bound was calculated without accounting for relative orbit perturbations. Therefore the achieved reconfiguration in a simulation environment that includes all orbit perturbations provided by \(S^3\) with the actual required delta-\(v\) of 0.5070 [m/s] is a very good result. Furthermore, figure 8.2 indicates the that both spacecraft require the same amount of delta-\(v\) which results in an equal fuel consumption. The deputy spacecraft requires 2.2\% more delta-\(v\) than the chief spacecraft.

The thruster levels can be seen in figure 8.3. The red dashed line indicates the desired thrust peak level of 26\(\mu\)N. One can see that the constraint is satisfied with minor and acceptable violations for the normal thrust applied by the deputy spacecraft at orbits 23 and 24. It is noticeable that the thrust level drops just before the out-of-plane reconfiguration finishes (orbit 19) and at the end of the in-plane reconfiguration as well (orbit 30). This is due to the small remaining errors, remember that the thrust level is proportional to the tracking error \(\Delta \delta a_u\) (see section 6.2). Figure 8.5 provides a closer view to the thrust inputs shown in figure 8.3. The \(\cos^N(u)\) is recognizable and also the peak of the control inputs at the desired peak level. For this simulation \(N = 14\) was chosen.

Figure 8.4 shows the mean quasi-nonsingular ROE error. This figure shows how first the out-of-plane variables \(\delta i_x\) and \(\delta i_y\) are controlled into the desired configuration, while the eccentricity vector rotates due to the differential Earth's oblateness effect. The inclination vector is controlled to a stable configuration with \(\delta i_x = 0\) such that once the desired configuration is achieved it will remain stable. Furthermore, one can see how the along-track separation \(\delta \lambda\) is successfully controlled by leveraging Keplerian dynamics to the desired reference. The overshoot of 17.1 m of \(\Delta \delta \lambda\) corresponds to 3.4\%.

The whole reconfiguration can be seen in figure 8.6. The yellow line corresponds to the path of the inclination vector, the red line top the taken path of the eccentricity vector and the blue line corresponds to the path of the differential semi-major axis and the along-track separation. One can see how the inclination vector is reconfigured on a straight path between desired and initial configuration. This behavior is desired as the length of the path taken in that graphical representation is proportional to the required delta-\(v\). During the initial period at which the inclination is controlled one can see how the eccentricity...
8.1. Thrust Constraint and Optimality

Figure 8.1: Relative position of the deputy spacecraft with respect to the chief spacecraft shown in the RTN frame.

Figure 8.2: Delta-\(v\) consumption of deputy and chief spacecraft given in [mm/s].

Figure 8.3: Thrust output levels and desired thrust level.

Figure 8.4: Mean quasi-nonsingular relative orbit element error given in [m].

vector rotates freely clockwise from (400[m], 400[m]) to (429[m], 368[m]). After orbit 20 (see figure 8.4 to compare) the out-of-plane has finished and the in-plane reconfiguration starts. It can be seen how the eccentricity vector is now controlled on a straight path to the desired reference. It is noticeable that this path experiences small perturbations. One can see how the the along-track separation (horizontal drift of the blue line) is passively controlled by controlling the difference of the semi-major axes \(\delta a\) (vertical drift of the blue line).
Figure 8.5: Display of the $\cos^N(u)$ shape of the control inputs. Here the control inputs into the normal direction of the RTN frame of the deputy (red) and the chief (purple) are shown.

Figure 8.6: Relative orbit element state-space. Three two-dimensional vectors (inclination vector(yellow), eccentricity vector (red), $[\delta \lambda, \delta a]$-vector (yellow)) are shown and together describe the complete six-dimensional state.
8.2. Reconfiguration with Wall Constraints

This example investigates the reconfiguration under imposed wall constraints on both $\delta e_x$ and $\delta i_y$. The absolute orbit parameters of the chief spacecraft, the initial and the desired relative configuration are the same as in example 8.1. The x-component of the eccentricity vector is constrained to be always larger than 300 m. This constraint is formulated by

$$c_1^T a_e \delta \alpha \geq d_1,$$

$$ \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} a_e \delta \alpha \geq 300 \text{ m}. \quad (8.9)$$

The y-component of the inclination vector is constrained to be always larger than 400 m, which is formulated by

$$c_2^T a_e \delta \alpha \geq d_2,$$

$$ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} a_e \delta \alpha \geq 400 \text{ m}. \quad (8.11)$$

Figure 8.7 shows the reconfiguration in ROE state-space. One can see how the both the inclination vector (yellow line) and the eccentricity vector (red line) first are directed straight into the direction of the reference. A rotation of the eccentricity vector is noticeable in the beginning - clockwise drift of red line at initial position. This is due to the differential Earth’s oblateness perturbation and due to the fact that first the inclination vector is controlled while the eccentricity vector remains uncontrolled during that time period. Please note, that this restriction (only tangential or normal thrust) is derived from the mission design of NetSat. NetSat is designed to provide only uni-directional thrust. The controller is capable of controlling both the eccentricity and inclination vectors simultaneously. Very close to the defined wall constraints both vectors start deflecting and one could argue that this deflection should take place well before the boundary. This can easily be achieved by defining a larger influence zone $\zeta$. For this simulation $\zeta = 4$ m was chosen. Usually a larger influence region of $\zeta = 100$ m is chosen to get a better and more fuel-efficient approach to the constraint boundary. But the intention of this example is to show how the vectors drift along the constraint boundaries while minimizing the error to the desired reference.
Figure 8.7: Reconfiguration shown in ROE state-space with two wall constraints.
Table 8.1: Continuous Low-Thrust formation keeping compared to flight results of TanDEM-X Autonomous Formation Flying experiment.

<table>
<thead>
<tr>
<th>Experiment/Simulation</th>
<th>Relative control error [m] (RMS/max)</th>
<th>delta-v/orbit [mm/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAFF</td>
<td>1.3/3.7 T:3.5/13.4</td>
<td>0.870</td>
</tr>
<tr>
<td>Low-thrust using true state</td>
<td>1.2/2.0 T:3.8/6.8</td>
<td>0.865</td>
</tr>
<tr>
<td>Low-thrust using estimated state</td>
<td>1.2/2.9 T:4.3/13.5</td>
<td>0.943</td>
</tr>
</tbody>
</table>

8.3. Formation Maintenance (TAFF)

This example shows the formation keeping capabilities of the stabilizing control utilizing the feedback gain matrix showed in equation 4.4. The TanDEM-X Autonomous Formation Flying (TAFF) [1] experiment is simulated using the full perturbation model provided by $S^3$ and fine integration steps of 0.5s. The experiment is to maintain a phasing orbit with the following desired reference given in meters:

$$a\delta \vec{a}_{0,r} = \begin{pmatrix} 0 \\ -123 \\ 68 \\ 387 \\ 0 \\ -449 \end{pmatrix} [\text{m}] \quad (8.12)$$

Continuous formation maintenance performance is highly dependent on accurate state estimation since estimation errors are of the same order of magnitude as tracking errors. Therefore, in order to reflect the performance of the designed controller two simulations are performed based on the true and the estimated state.

The TAFF experiment only maintained the in-plane motion of the formation in order to keep a single configuration of the spacecraft. For that reason, in this simulation the thrust in cross-track direction is set to zero to obtain comparable results. TAFF’s formation maintenance experiment with the highest accuracy is used as a comparison here. Control was applied twice each three orbits. For the Synthetic Aperture Radar experiments it is important that no thrust is applied during the experiments. Thus, this comparison is only used to validate the formation capabilities and the delta-v consumption of the continuous low-thrust control strategy.

Figures 8.8 to 8.11 show the formation keeping performance using the the estimated relative state as input for the controller. One can see that the eccentricity vector has a constant offset and all other all orbit elements have a zero mean error offset. The thrust stays within its boundaries $F \leq 80 \mu N$ and is equally distributed between both spacecraft (see delta-v consumption). Thrust inputs to re-orient the eccentricity vector are separated by half a orbit periods as given by the intelligent feedback gain matrix.
8.4. Time Constraint

In this example a reconfiguration is performed with an imposed time constraint of 35 orbits reconfiguration time. The initial and final relative states are given by

\[
a_c \delta \alpha_0 = \begin{pmatrix} a_c \delta a \\ a_c \delta \lambda \\ a_c \delta e_x \\ a_c \delta e_y \\ a_c \delta i_x \\ a_c \delta i_y \end{pmatrix} = \begin{pmatrix} 0 \\ 612 \\ 400 \\ 400 \\ 100 \\ 500 \end{pmatrix} [\text{m}], \quad a_c \delta \alpha_{\text{ref}} = \begin{pmatrix} a_c \delta a \\ a_c \delta \lambda \\ a_c \delta e_x \\ a_c \delta e_y \\ a_c \delta i_x \\ a_c \delta i_y \end{pmatrix}_{\text{ref}} = \begin{pmatrix} 0 \\ 212 \\ 200 \\ 200 \\ 0 \\ 200 \end{pmatrix} [\text{m}] .
\]  

Figure 8.12 shows the relative orbit element error over time. During the first orbit period no control inputs are applied in order to estimate the semi-major axis drift due to atmospheric drift. Therefore the reconfiguration has to be finished after 36 instead of 35 orbits. Equations (5.9) and (5.10) are online to calculate number of orbits required for the in-plane
8.4. Time Constraint

Figure 8.12: Relative orbit element errors shown over time. The left black dashed line indicates the start of the out-of-plane reconfiguration. The right black dashed line indicates the start of the in-plane reconfiguration. The red dashed line is located at 36 orbits and represents the time constraint.

and the out-of-plane reconfiguration so to yield the same thrust level that has to be applied into the tangential and normal direction, respectively, as shown by

\[
\#\text{orbits}_{ip} = \frac{\|\Delta\delta e\|}{2(1 - e_c)\|\Delta\delta \iota\| + \|\Delta\delta e\|} \cdot \#\text{orbits},
\]

\[
= \frac{\sqrt{(200 \text{ [m]})^2 + (200 \text{ [m]})^2}}{2(1 - 0.00013)\sqrt{(100 \text{ [m]})^2 + (300 \text{ [m]})^2} + \sqrt{(200 \text{ [m]})^2 + (200 \text{ [m]})^2}} \cdot 35,
\]

\[
\approx 11,
\]

\[
\#\text{orbits}_{oop} = \frac{2(1 - e_c)\|\Delta\delta \iota\|}{2(1 - e_c)\|\Delta\delta \iota\| + \|\Delta\delta e\|} \cdot \#\text{orbits},
\]

\[
= \frac{2(1 - 0.00013)\sqrt{(100 \text{ [m]})^2 + (300 \text{ [m]})^2}}{2(1 - 0.00013)\sqrt{(100 \text{ [m]})^2 + (300 \text{ [m]})^2} + \sqrt{(200 \text{ [m]})^2 + (200 \text{ [m]})^2}} \cdot 35,
\]

\[
\approx 24.
\]

The actual thrust levels can be seen in figure 8.13. The thrust levels for the in-plane reconfiguration (blue and yellow lines) are of the same magnitude as the thrust levels for the out-of-plane reconfiguration (red and purple lines). There are three thrust peaks noticeable near the end of the in-plane reconfiguration. This behavior occurred when the along-track separation error \(\delta \lambda\) is controlled to zero and the applied reference for the difference in the semi-major axis \(\delta a\) changes too quickly. Further tuning of the \(\tau\) parameter in equation (4.11) is required. A larger \(\tau\) would prevent the thrust peaks, but on the other hand it would take longer until \(\Delta\delta \lambda\) is controlled to zero. Overall, figure 8.12 shows that the reconfigura-
The reconfiguration in fact is performed in 35 orbits. This figure is also suitable to see the effect of differential J2 onto the relative state as seen in the ROE state-space. The most significant effect of the differential J2 perturbation is the rotation of the eccentricity vector which is represented by the yellow and purple lines. During the first period in which the out-of-plane reconfiguration takes place (between the two black dashed lines) one can see how the x-component of the eccentricity vector increases while the y-component decreases (rotation). As soon as the in-plane reconfiguration starts the eccentricity vector is controlled to the desired reference.
Conclusion and Way Forward

This thesis enables satellite-formation flying for satellites equipped with continuous low-thrust propulsion systems. The proposed control and guidance strategy are entirely autonomous and impose only minor demands on the attitude control system. Thrust is only applied into the tangential and normal direction of the co-rotating orbital frame. Thus, no thrust profiles have to be followed by the attitude control system. Furthermore, analytic solutions for the calculation of fuel-optimal thrust locations allow to run the code online. A simulation environment in MATLAB/Simulink is provided with example code and a Simulink library for formation-flying which augments the high-fidelity S3 propagator developed by Duncan Eddy at Stanford University. It has been shown that the continuous control strategy performs near-optimal as compared to the fuel-optimal impulsive solution with an increase in fuel consumption of about 4%. The discussed constraints can be enforced using the proposed reference governor.

Next steps include the extension of this work to swarms of satellites. Problems will arise for the fuel balancing of all spacecraft involved in the swarm. Ideas in order to solve this problems are to 1) assign pairs of spacecraft in a swarm formation that will apply the proposed control strategy of this thesis in order to achieve the same fuel consumption and to 2) find an autonomous on-board algorithm that is smart in finding a virtual center of the formation to which all other spacecraft pairs keep or reconfigure their states. This way it is possible to find a virtual center that ensures equal fuel consumption between already balanced spacecraft pairs. Furthermore, a repulsive potential field may be imposed on each spacecraft for active collision avoidance to move forward the passive collision approach shown in this thesis. In the proposed control strategy only Keplerian dynamics are leveraged in order to achieve near-optimal fuel consumption. The way forward is to leverage the differential J2 effect as well if time constraints allow. The full rotation of the eccentricity vector due to differential J2 occurs in about one month of time. If the rotation is beneficial for a reconfiguration it is wise to wait and use this perturbation. An algorithm has to be developed to do this autonomously.
Bibliography


Matrices

Plant matrix for the differential Earth oblateness effect is given by \[10\]

\[
\hat{A}_{J2}(\alpha_c) = \kappa \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{7}{2} E P & 0 & e_y G F P & e_y G F P & -F S & 0 & 0 \\
\frac{7}{2} e_y Q & 0 & -4 e_x e_y G Q & -(1 + 4 e_y^2 G) Q & 5 e_y S & 0 & 0 \\
-\frac{7}{2} e_x Q & 0 & (1 + 4 e_x^2 G) Q & 4 e_x e_y G Q & -5 e_x S & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{7}{2} S & 0 & -4 e_x G S & -4 e_y G S & 2 T & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\] (A.1)

The plant matrix for the differential Earth oblateness effect for the reduced model is given by

\[
A_{J2}(\alpha_c) = \kappa \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{7}{2} e_y Q & -4 e_x e_y G Q & -(1 + 4 e_y^2 G) Q & 5 e_y S & 0 & 0 \\
-\frac{7}{2} e_x Q & (1 + 4 e_x^2 G) Q & 4 e_x e_y G Q & -5 e_x S & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{7}{2} S & -4 e_x G S & -4 e_y G S & 2 T & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\] (A.2)

\[
\dot{\omega} = \kappa Q; \quad \gamma = \frac{3}{4} J_2 R^2 \sqrt{\mu}; \quad \eta = \sqrt{1 - e^2}; \quad \kappa = \frac{\gamma}{a_c^{7/2} \eta_c^4} \\
e_x = e_c \cos(\omega_c); \quad e_y = e_c \sin(\omega_c); \quad E = 1 + \eta_c; \quad F = 4 + 3 \eta_c; \quad G = \frac{1}{\eta_c^2} \\
P = 3 \cos(i_c)^2 - 1; \quad Q = 5 \cos(i_c)^2 - 1; \quad S = \sin(2i_c); \quad T = \sin(i_c)^2
\]

Differential drag plant matrix \[10\]:

\[
\hat{A}_{\text{drag}} = \begin{pmatrix}
0^{6 \times 6} & 1^{6 \times 1} \\
0^{5 \times 1} & 0^{1 \times 7}
\end{pmatrix}
\] (A.3)

The differential drag plant matrix for the reduced state is given by
\[ \mathbf{A}_{\text{drag}} = \begin{bmatrix} \mathbf{0}_{5 \times 5} & 1 \\ \mathbf{0}_{4 \times 1} & \mathbf{0}_{1 \times 6} \end{bmatrix} \]  
(A.4)

The plant matrix for relative Keplerian dynamics is given by
\[ \mathbf{A}_{\text{Kepler}} = \begin{bmatrix} 0 \\ -1.5n \\ \mathbf{0}_{5 \times 1} \end{bmatrix} \]  
(A.5)

where \( n \) is the mean motion. The plant matrix describing the relative Keplerian motion for the reduced model is given by
\[ \mathbf{A}_{\text{Kepler}} = \mathbf{0}^{6 \times 6} \]  
(A.6)

Control input matrix for quasi-nonsingular ROE:

\[
\mathbf{B}(\alpha_c) = \frac{1}{an} \begin{bmatrix} \frac{2}{\eta} e \sin f & \frac{2}{\eta} (1 + e \cos f) & 0 \\ \frac{2\eta^2}{1 + e \cos f} & 0 & 0 \\ \eta \sin \theta & \eta \left( \frac{2 + e \cos f}{1 + e \cos f} \right) \cos \theta + e_x & \frac{\eta e_y}{\tan \eta} \sin \theta \\ -\eta \cos \theta & \eta \left( \frac{2 + e \cos f}{1 + e \cos f} \right) \sin \theta + e_y & \frac{-\eta e_x}{\tan \eta} \sin \theta \\ 0 & 0 & \eta \cos \theta \frac{1}{1 + e \cos f} \\ 0 & 0 & \eta \sin \theta \frac{1}{1 + e \cos f} \end{bmatrix}
\]  
(A.7)

Control input matrix for quasi-nonsingular ROE of the reduced model (used in the reference governor):

\[
\mathbf{B}(\alpha_c) = \frac{1}{an} \begin{bmatrix} \frac{2}{\eta} (1 + e \cos f) & 0 \\ \eta \left( \frac{2 + e \cos f}{1 + e \cos f} \right) \cos \theta + e_x & \frac{\eta e_y}{\tan \eta} \sin \theta \\ \eta \left( \frac{2 + e \cos f}{1 + e \cos f} \right) \sin \theta + e_y & \frac{-\eta e_x}{\tan \eta} \sin \theta \\ 0 & \eta \cos \theta \frac{1}{1 + e \cos f} \\ 0 & \eta \sin \theta \frac{1}{1 + e \cos f} \end{bmatrix}
\]  
(A.8)

where \( f \) is the true anomaly and \( \theta = f + \omega \).
\[
\frac{\partial \delta r_{\min}^{\text{min}}}{\partial \delta e_x} = \frac{a}{2\sqrt{2}} \left( \delta e^2 + \delta i^2 - \sqrt{\delta e^4 + \delta i^4 - 2\delta e^2 \delta i^2 \cos(2(\varphi - \theta))} \right)^{-1/2} \left[ 2\delta e_x \\
- \frac{1}{2} (\delta e^4 + \delta i^4 - 2\delta e^2 \delta i^2 \cos(2(\varphi - \theta)))^{-1/2} \left( 4\delta e^2 \delta e_x \right. \\
- 4\delta e_x \delta i^2 \cos(2(\varphi - \theta)) - 4\delta i^2 \delta e_x \sin(2(\varphi - \theta)) \right] \tag{A.9}
\]

\[
\frac{\partial \delta r_{\min}^{\text{min}}}{\partial \delta e_y} = \frac{a}{2\sqrt{2}} \left( \delta e^2 + \delta i^2 - \sqrt{\delta e^4 + \delta i^4 - 2\delta e^2 \delta i^2 \cos(2(\varphi - \theta))} \right)^{-1/2} \left[ 2\delta e_y \\
- \frac{1}{2} (\delta e^4 + \delta i^4 - 2\delta e^2 \delta i^2 \cos(2(\varphi - \theta)))^{-1/2} \left( 4\delta e^2 \delta e_y \right. \\
- 4\delta e_y \delta i^2 \cos(2(\varphi - \theta)) + 4\delta i^2 \delta e_y \sin(2(\varphi - \theta)) \right] \tag{A.10}
\]

\[
\frac{\partial \delta r_{\min}^{\text{min}}}{\partial \delta i_x} = \frac{a}{2\sqrt{2}} \left( \delta e^2 + \delta i^2 - \sqrt{\delta e^4 + \delta i^4 - 2\delta e^2 \delta i^2 \cos(2(\varphi - \theta))} \right)^{-1/2} \left[ 2\delta i_x \\
- \frac{1}{2} (\delta e^4 + \delta i^4 - 2\delta e^2 \delta i^2 \cos(2(\varphi - \theta)))^{-1/2} \left( 4\delta i^2 \delta i_x \right. \\
- 4\delta i_x \delta e^2 \cos(2(\varphi - \theta)) + 4\delta e^2 \delta i_x \sin(2(\varphi - \theta)) \right] \tag{A.11}
\]

\[
\frac{\partial \delta r_{\min}^{\text{min}}}{\partial \delta i_y} = \frac{a}{2\sqrt{2}} \left( \delta e^2 + \delta i^2 - \sqrt{\delta e^4 + \delta i^4 - 2\delta e^2 \delta i^2 \cos(2(\varphi - \theta))} \right)^{-1/2} \left[ 2\delta i_y \\
- \frac{1}{2} (\delta e^4 + \delta i^4 - 2\delta e^2 \delta i^2 \cos(2(\varphi - \theta)))^{-1/2} \left( 4\delta i^2 \delta i_y \right. \\
- 4\delta i_y \delta e^2 \cos(2(\varphi - \theta)) - 4\delta e^2 \delta i_y \sin(2(\varphi - \theta)) \right] \tag{A.12}
\]
Derivation of the Out-of-Plane Delta-v Lower Bound

The change of the inclination vector as a result of a maneuver into the normal direction of the RTN frame is described by the Gauss Variational Equations given in equation (A.7) as

\[ \|\Delta \delta i\| = \left[ |\Delta \delta i_x|^2 + |\Delta \delta i_y|^2 \right]^{0.5}, \]

\[ \|\Delta \delta i\| = \left[ \left( \frac{\eta}{a_c n} \frac{\cos(\theta)}{1 + e_c \cos f} \Delta v_n \right)^2 + \left( \frac{\eta}{a_c n} \frac{\sin(\theta)}{1 + e_c \cos f} \Delta v_n \right)^2 \right]^{0.5}, \]

\[ \|\Delta \delta i\| = \left[ \frac{\eta^2 \Delta v_n^2}{a_c n^2 (1 + e \cos f)^2} \left( \cos^2 \theta + \sin^2 \theta \right) \right]^{0.5}, \]

\[ \|\Delta \delta i\| = \frac{\eta \Delta v_n}{a_c n (1 + e \cos f)}. \]

Solving equation (B.4) for \( \Delta v_n \) and renaming \( \Delta v_n = \Delta v_{i_p} \) yields

\[ \Delta v_{i_p} = \frac{a_c n (1 + e \cos f)}{\eta} \|\Delta \delta i\|, \]

which is minimized for \( f = \pm \pi \)

\[ \Delta v_{i_p}^{\text{opt}} = \frac{a_c n (1 - e)}{\eta} \|\Delta \delta i\|. \]