Robust Control of Large-Scale Satellite Constellations Using System-Level Synthesis

F.J.P. Ballast

Master of Science Thesis

TU Delft

Delft Center for Systems and Control
Cognitive Robotics

Robust Control of Large-Scale Satellite Constellations Using System-Level Synthesis

Master of Science Thesis

For the degrees of Master of Science in Systems and Control and in Robotics at Delft University of Technology

F.J.P. Ballast

December 6, 2023

Faculty of Mechanical, Maritime and Materials Engineering (3mE)
Delft University of Technology

Abstract

The ‘New Space’ mentality is gaining in popularity and is at the basis of the growing size of satellite constellations. These satellite constellations are used for technologies such as satellite navigation and internet, but a clear framework to controlling large satellite constellations is missing. Therefore, a new approach is developed to work efficiently with thousands of satellites by finding a suitable model, controller and problem formulation.

A new model is developed that is linear time-invariant, fully based on physics and can include J_2 perturbations. Furthermore, it can work well for both in-plane and out-of-plane movements. This model is paired with the lumped System-Level Synthesis control framework: a robust control algorithm that optimises the closed-loop transfer function. Due to a modification, the applied controller is less conservative than the original controller and faster than other modifications. The problem formulation is rewritten to a standard quadratic problem to significantly increase the rate at which these problems can be solved. This includes rewriting one-norms and infinity norms, but also a new formulation for these System-Level Synthesis problems in general.

These findings are tested in a simulation of over two hundred satellites, where the satellites are controlled with the new model, the robust controller, the new problem formulation and collision avoidance constraint. These collision avoidance constraints are added to ensure that satellites keep a safe distance between themselves at all times. This includes constraints between satellites within the same plane, but also between satellites that cross each other’s orbit while close to each other.

Preface

This document is the final part of my Master of Science graduation thesis, which consists of two main parts. The first, completed in February, is a literature survey consisting of an overview of the relevant literature that is already available. The second part, which is this report, is the actual thesis itself. Here, the knowledge obtained during the literature survey is then applied to the problem, along with several new contributions to extend the scientific domain further.

The topic of this thesis, "*Robust Control of Large-Scale Satellite Constellations Using System-Level Synthesis*", covers several different subjects, such as orbital mechanics, control theory and mathematical optimisation. Where the latter two have made an appearance during the MSc. Systems & Control and the MSc. Robotics, orbital mechanics was a topic in which I was a novice. I was interested in dynamics, and many hours later, I eventually developed an understanding of orbital mechanics. The process has been challenging, but these challenges make the results more rewarding at the same time.

Acknowledgements

First and foremost, I would like to thank my supervisors prof. dr. ir. Tamás Keviczky and dr. ir. Laura Ferranti for their assistance during the writing of this master thesis. Their guidance helped structure the year-long process leading to this thesis and provided crucial feedback on the report's content.

Second, I would like to thank dr. ir. Samir Bennani and dr. Valentin Preda from ESA for their help during the early stages of the project. Their expertise in space missions and familiarity with the literature helped guide the project to a realistic but valuable thesis.

Finally, I want to thank everyone who supported me during the thesis, both for the kind and inspiring words and for the joined study sessions. Your support was invaluable through the difficult times.

Delft University of Technology F.J.P. Ballast
December 6, 2023

“To infinity and beyond.”

— *Buzz Lightyear*

Table of Contents

List of Figures

List of Tables

List of Acronyms

List of Symbols

Bold Symbols

$\boldsymbol{\delta}_x$	Signal of the disturbance on the state
$\boldsymbol{\eta}$	Lumped uncertainty signal
$\boldsymbol{\Sigma}$	Upper bound on $\boldsymbol{\eta}$
$\boldsymbol{\varphi}_u^t$	Vector equivalent of $\boldsymbol{\Phi}_u^t$
$\boldsymbol{\varphi}_x^t$	Vector equivalent of $\boldsymbol{\Phi}_x^t$
$\boldsymbol{\Phi}_u^0$	First block column of $\boldsymbol{\Phi}_u$
$\boldsymbol{\Phi}_u$	State-feedback transfer matrix from $\boldsymbol{\delta}_x$ to $\mathbf{u}$
$\boldsymbol{\Phi}_x^0$	First block column of $\boldsymbol{\Phi}_x$
$\boldsymbol{\Phi}_x$	State-feedback transfer matrix from $\boldsymbol{\delta}_x$ to $\mathbf{x}$
$\mathbf{d}$	Any perturbing force not present in the Keplerian two-body problem
$\mathbf{K}$	Causal linear time-varying (LTV) state-feedback controller
$\mathbf{r}$	Position of the satellite in the ECI frame
$\mathbf{u}$	Control input signal
$\mathbf{v}$	First derivative of r with respect to time t
$\mathbf{w}$	Disturbance signal
$\mathbf{x}$	State signal

Greek Symbols

$\epsilon_{\mathcal{A}}$	Uncertainty bound on $\mathcal{A}$
$\epsilon_{\mathcal{B}_2}$	Uncertainty bound on $\mathcal{B}_2$
η_t	Lumped uncertainty
η	Ratio between semi-minor axis and semi-major axis
μ	Gravitational parameter of the Earth
Ω	Right Ascension of the Ascending Node (RAAN)
ω	Argument of periapsis
$\Phi_u^{i,j}$	Block in $\boldsymbol{\Phi}_u$
$\Phi_x^{i,j}$	Block in $\boldsymbol{\Phi}_x$
σ_t	Upper bound on η_t
θ	Argument of latitude
υ	Mean argument of latitude
$\varphi_u^{i,j}$	Vector equivalent of $\Phi_u^{i,j}$
$\varphi_x^{i,j}$	Vector equivalent of $\Phi_x^{i,j}$

Latin Symbols

a	Semi-major axis
b	Semi-minor axis
e_j	Standard basis 'j'
E	Eccentric anomaly
e	Eccentricty
f	True anomaly
H_x	Matrix for the constraint $H_x x \leq h_x$
h_x	Vector for the constraint $H_x x \leq h_x$
h	Angular momentum per unit mass
i_{sp}	Indices for which the sparse matrix is non-zero
i	Inclination angle
J_k	Zonal coefficient k
M_0	Mean anomaly at epoch
M	Mean anomaly
N_u^j	Link between the one-norm of $\Phi_u^{t,t-i,j}$ and $\varphi_u^{t,t-i}$
N_x^j	Link between the one-norm of $\Phi_x^{t,t-i,j}$ and $\varphi_x^{t,t-i}$
n	Mean motion
p	Semi-latus rectum
R_e	Equatorial radius of the Earth
t_0	Time at epoch
T	Time horizon
t	Time
u	Control input as acceleration
Z	The block-downshift operator

Subscripts and Superscripts

$(\cdot)^{\varphi}$	Parameter corresponding to vectorised SLS problem
$(\cdot)_{\text{coll}}$	Collision angle
$(\cdot)_c$	Chief spacecraft
$(\cdot)_d$	Deputy spacecraft
$(\cdot)_n$	Component in normal direction
$(\cdot)_r$	Component in radial direction
$(\cdot)_t$	Component in tangential direction
$\bar{(\cdot)}$	Averaged component
$\ddot{(\cdot)}$	Second derivative with respect to time
$\dot{(\cdot)}$	First derivative with respect to time
$\hat{(\cdot)}$	Parameter for estimated dynamics
$\tilde{(\cdot)}$	Parameter for lumped uncertainty problem

Number Sets

$\mathbb{R}$	Set of all real numbers
$\mathbb{Z}_i^j$	Set of all integer numbers from i up to and including j

Calligraphic Symbols

$\mathcal{A}$	Block diagonal matrix of A_t for $t \in \mathbb{Z}_0^{T-1}$
$\mathcal{B}_2$	Block diagonal matrix of $B_{2,t}$ for $t \in \mathbb{Z}_0^{T-1}$
$\mathcal{C}$	Set for collision avoidance constraints
$\mathcal{D}$	Convex set of possible disturbances
$\mathcal{K}_u$	Link between x_0 and $\boldsymbol{\varphi}_u^0$
$\mathcal{K}_x$	Link between x_0 and $\boldsymbol{\varphi}_x^0$
$\mathcal{O}(\cdot)$	Order of magnitude of $(\cdot)$
$\mathcal{Q}$	Weight on the states for finite horizon SLS problem
$\mathcal{R}$	Perturbing potential where a common choice are zonal gravitational harmonics
$\mathcal{R}$	Weight on the inputs for finite horizon SLS problem
$\mathcal{S}$	Convex set of constraints on $\mathbf{\Phi}_x$ or $\mathbf{\Phi}_u$
$\mathcal{U}$	Convex set of input constraints
$\mathcal{X}$	Convex set of state constraints

Different Symbols

$\mathfrak{P}$	Poisson matrix
œ	Orbital Elements

Chapter 1

Introduction

This chapter provides an introduction to the thesis work. As no assumptions are made on the level of knowledge of the reader in the space domain, an introduction to modern space missions is provided in Section 1-1 first, followed by a discussion of different types of multi-spacecraft missions in Section 1-2. An overview of (the history of) satellite constellations is provided in Section 1-3, after which the current state is discussed in Section 1-4. Problems arising from this state are then discussed in Section 1-5. Finally, an outline for the rest of the work and an overview of the contributions in this work are presented in Sections 1-6 and 1-7.

1-1 Introduction To Modern Space Missions

Traditionally, space missions were often carried out using a single or a handful of spacecraft. Missions for satellite navigation, which usually contain approximately 24 to 30 satellites, were on the larger end of multi-spacecraft missions. However, recent developments have shifted the industry to the 'New Space' or 'Space 4.0' mentality [1], [2], which is built from three main aspects:

1. Space privatisation. Where traditionally government-funded companies such as NASA, ESA and Roskosmos were the main operators in space, a rising part of the space industry is in private hands nowadays. Companies such as SpaceX, Blue Origin, Virgin Galactic and Planet Labs are public companies with the goal to sell commercial services.

2. Satellite miniaturisation. As a continuation of the "Microspace" movement from the 1990s and early 2000s [2], the spacecraft themselves are becoming smaller, cheaper and easier to manufacture. The interest in smaller spacecraft has led to the design of CubeSats, which are small and lightweight cube-like satellites that have, for example, been employed by Planet Labs [3].

3. Novel services based on space data. Combining both miniaturisation and privatisation, it is now relatively inexpensive and fast to launch a small spacecraft into space. This has led to many new or improved services, such as Earth observations, radio frequency monitoring, asset tracking and sensor data collection [1]. Satellite internet has also gained popularity, as even though it has been around for two decades, it was regarded as an expensive and slow option with low capacity [2]. With a larger number of satellites, it is possible to decrease the orbit's altitude while still providing continuous services, decreasing the latency in the network.

1-2 Classification of Multi-Spacecraft Missions

The ‘New Space’ movement has increased the number of multi-spacecraft missions, which fall into four categories depending on the inter-satellite distance and the required control accuracy. This is depicted in Fig. 1-1, where four types of missions are considered: rendezvous & docking, formation flying, constellations and swarms.

Rendezvous & docking missions entail, as the name suggests, that two (or multiple) spacecraft meet and connect with a physical connection (i.e., the docking). These missions have been the most common in the past, of which many missions with the International Space Station (ISS) are a clear example. The rendezvous part requires a small inter-satellite distance, and the docking requires a high control accuracy.

Formation flying missions were among the first missions executed in space, with the capsules Gemini 6A and Gemini 7 in 1965 being a prime example [4]. It is characterised by a medium-sized inter-satellite distance and control accuracy, and usually consists of only a handful of (possibly heterogeneous) satellites. A more modern example of a formation flying mission is the Swedish Prisma mission, consisting of two heterogeneous satellites launched in 2010 [4].

Constellations span a global area and usually involve multiple spacecraft in one or multiple orbits around a celestial body. Constellations can be used to make observations of that celestial body, make measurements or create a service such as satellite internet or Global Positioning System (GPS). The latter is one of the oldest examples of satellite navigation, which consists of 24 satellites and has been operational for almost three decades.

Swarms are the newest type of mission and consist of several tens to several thousands of satellites. Swarms are, as opposed to the rendezvous & docking and formation flying missions, mostly made up of homogeneous satellites. Because swarms are relatively new and span a wide region of different missions in Fig. 1-1, it can be hard to distinguish between a swarm and a different type of mission. For example, a mission such as the Orbiting Low Frequency Array (OLFAR) [5] is deemed to be a swarm by some authors [6] but considered to be a constellation by others [7]. This mission aims at receiving radio signals from 30 kHz up to 40 MHz, and the aperture diameter of the radar array can be up to 100 km, which would be impossible with a single spacecraft.

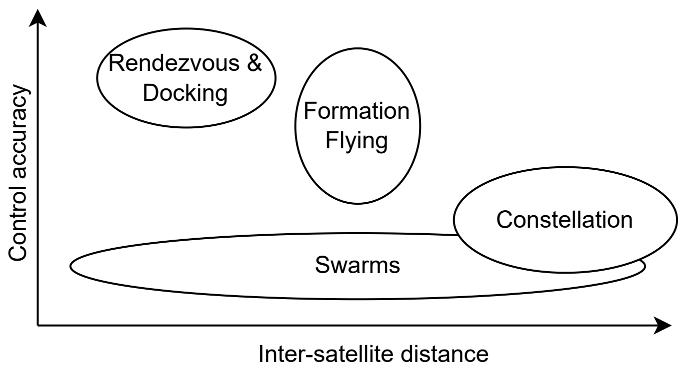


Figure 1-1: Classification of different types of multi-spacecraft missions adapted from [8].

1-3 Constellations For Satellite Internet

Of these multi-spacecraft missions, the large-scale missions with constellations and swarms have been the most affected by the rising popularity of the 'New Space' mentality. Especially those missions benefit from a larger scale and allow for the novel services discussed in Section 1-1, such as Earth imaging, radio frequency monitoring and faster satellite internet.

Especially the latter has many companies competing, as they foresee a large commercial market for satellite internet. However, as already briefly mentioned in Section 1-1, satellite internet is not a new technology, and this service has been provided for over two decades. These satellites were usually placed in a Geostationary Earth Orbit (GEO) of 36.000 km, which leads to propagation delays of roughly 120 ms [1]. This is significantly higher than the propagation delays for Very Low Earth Orbit (VLEO) (roughly 50-85 µs) and Low Earth Orbit (LEO) (roughly 2 ms) satellites. This latency advantage also led to many companies filing for LEO constellations as early as the 1990s, but these projects failed mostly due to the expensive technology [9]. The disadvantage of these lower orbits is that more satellites are required to provide continuous services, which was not cost-effective at that time.

Nevertheless, many companies are working on providing high-speed satellite internet now, with SpaceX (Starlink), Amazon (Kuiper), OneWeb and Telesat among the most well-known [9]. Satellite internet can enhance network reliability, reach unserved areas such as deserts, oceans and forests, and scale their service to offload the terrestrial network [1]. The maritime and aviation industries are clear examples of industries that cannot always use a ground network.

The four previously mentioned providers of satellite internet are compared in [9]. The planned altitude, number of planes, satellites per plane and the inclination of the constellation are shown in Table 1-1. The inclination corresponds to the angle the orbits make with the equatorial plane of the Earth, and most of the satellites are placed into orbits with an inclination between 40° and 55°, as this corresponds to the most densely populated areas.

Table 1-1: Constellation characteristics of four systems [9].

Company	Altitude	Inclination	Planes	Satellites per plane	Number of satellites
Telesat	1.015 km	98.98°	27	13	1.617
	1.325 km	50.88°	40	33	
OneWeb	1.200 km	87.9°	36	49	6.372
	1.200 km	55°	32	72	
	1.200 km	40	32	72	
SpaceX	540 km	53.2°	72	22	4.408
	550 km	53°	72	22	
	560 km	97.6°	6	58	
	560 km	97.6°	4	43	
	570 km	70°	36	20	
Amazon	590 km	33°	28	28	3.236
	610 km	42°	36	36	
	630 km	51.9°	34	34	

1-4 Current State of Constellation Control

Due to the commercial possibilities at stake, most companies dealing with satellite constellations do not publicly share information on their approach. Although people have tried to analyse the work of these companies, such as estimating the control accuracy of a Starlink constellation [10], the approach of these companies remains mostly a black box. However, several papers have been published by one company, Planet Labs, on their approach, where algorithms have been applied and tested on real satellite constellations [3], [11]–[13].

Planet Labs is a California-based space company that designs, builds and operates large CubeSat constellations for Earth observation. Their satellites always point towards the Earth and continuously make images of the Earth. They use this data for commercial and humanitarian purposes [3].

An interesting detail of the Planet Labs satellites is that they have no propulsion system but instead control their position using differential drag. With differential drag, satellites decelerate by increasing their frontal area or drag coefficient and thus increasing the aerodynamic forces acting on the satellite. Satellites can control their along-track position along the orbit with differential drag. Differential drag control is less expensive and saves engineering and regulatory work, but is also relatively slow, can only be used in a LEO or VLEO and directly affects the lifetime of the satellite [11]. The latter follows from the fact that the increased drag forces also decrease the satellite's altitude. 'Fuel' optimal control is therefore important, as the operational lifetime of a satellite can be significantly reduced with inefficient control.

Planet Labs uses simulated annealing to control their constellation with a simple double integrator model. They can control 88 satellites using this algorithm, although only the angular error is controlled. The input for each satellite is either 1 (high drag) or 0 (low drag) and is randomly sampled. The algorithm is provided in Algorithm 1:

Algorithm 1 Command generator from Planet Labs.

Input: $k_{\max}, \theta_{i,\text{desired}}$
Output: u
  $\text{cost}_{\text{old}} \leftarrow \infty$
  for $k = 0; k < k_{\max}; k = k+1,$ **do**
    Create u_{new} with randomly flipped command u
    Predict future states $\theta_{i,t}$ under u_{new}
    Compute cost: $\text{cost}_{new} = \sum_{i=0}^{n_{\text{sats}}} \sum_{t=0}^{t_{\text{final}}} (\theta_{i,t} - \theta_{i,\text{desired}})^2$
    if $P(\text{cost}_{\text{old}}, \text{cost}_{\text{new}}, k, k_{\max}) \geq \text{random}(0,1)$, **then**
      $u \leftarrow u_{\text{new}}$
    end if
  end for

where the probability function $P(\text{cost}_{\text{old}}, \text{cost}_{\text{new}}, k, k_{\max})$ is defined as:

$$P(\text{cost}_{\text{old}}, \text{cost}_{\text{new}}, k, k_{\max}) := \begin{cases} 1, & \text{if } \text{cost}_{\text{new}} < \text{cost}_{\text{old}}, \\ \exp\left(\frac{\text{cost}_{\text{old}} - \text{cost}_{\text{new}}}{t_0\left(1 - \frac{k}{k_{\max}}\right)}\right), & \text{if } \text{cost}_{\text{new}} > \text{cost}_{\text{old}}. \end{cases}$$

1-5 Problem Statement

When the scale of a constellation changes from tens to thousands of satellites, a new approach is required or desired as the original approach breaks down at the new scale. For example, operations traditionally performed manually by skilled operators may need to be automated. Traditionally, the number of skilled personnel was larger than that of spacecraft, resulting in thousands of skilled operators for the constellations shown in Table 1-1 [14].

Furthermore, algorithms that work well for a handful of satellites can fall victim to the curse of dimensionality when the number of satellites grows. Where computations might take seconds or minutes for ten satellites, they can take hours or days for hundreds of satellites if the algorithm scales poorly. The approach of Planet Labs in Algorithm 1 is an example of this, as simulated annealing with randomly sampled control inputs does not scale well when the size of the constellations grows larger. As simulated annealing is a general nonlinear optimiser and no gradient information is used, optimisation is relatively slow. Other shortcomings of this approach include:

- Their double integrator model [11] only controls the angular error and requires separate controllers for the radial and out-of-plane errors. The former is coupled to the angular errors, which are ideally controlled closely together. They provide no instructions on how to do so in [11].
- Their model does not include, nor account for, any perturbations and is very simplistic. This allows for large modelling errors, while at the same time, it is not possible to guarantee that constraints are met. The space industry's costs are still very high, and collisions leading to space debris could compromise future missions. Therefore, a robust control approach is desired to guarantee that (collision avoidance) constraints are met.
- They have not included any constraints in their problem. Input constraints can easily be included by sampling u only from the feasible set, but this is not as simple for state constraints, as the future states are only considered after sampling the new u. If it is impossible to inversely calculate the set of possible inputs such that the states are within their allowed bounds, the number of sampled inputs needs to be increased, further decreasing the speed. Collision avoidance constraints can, due to the angle being the only state variable, only take on a very simple form.

To improve upon this issues, the goal of this research is to answer the following question:

> *How can a large constellation of several hundreds of satellites and multiple planes be robustly controlled while maintaining a safe distance between themselves?*

where the following sub-questions are used to answer this question:

- *What dynamical model best controls satellites for both in-plane and out-of-plane movements?*
- *What control algorithm can be used best to control a satellite robustly?*
- *How can a different formulation of the optimisation problem scale up the control algorithm?*
- *How can collision avoidance constraints be (robustly) formulated with this dynamical model and control algorithm?*

1-6 Outline

This report is divided into six chapters, where chapters Chapters 2 to 5 each answer one of the sub-questions from Section 1-5:

1. Chapter 1 provides an introduction to the problem at hand and the research done. It provides an overview of the report in general and the contributions that are claimed to arise from this work.
2. Chapter 2 introduces the reader to orbital mechanics and the derivations of different dynamical models. This includes models already present in the literature but also a new model derived in this work.
3. Chapter 3 gives an overview of the System-Level Synthesis (SLS) framework and why it can help control a large satellite constellation. This also includes a discussion of a robust SLS variant.
4. Chapter 4 discusses how the optimisation problem at hand can be solved in a reasonable time, also for large constellations. This includes a comparison of different solvers and problem formulations to reach the maximum potential of these solvers.
5. Chapter 5 shows several large-scale simulations with the results from the previous chapters combined. As the number of satellites increases significantly in this chapter, collision-avoidance constraints are also discussed.
6. Chapter 6 finalises the work with a conclusion of the results, a discussion of the work and possible future work.

1-7 Contributions

This report provides the following contributions:

- A new linear time-invariant (LTI) model has been developed that works well for both in-plane and out-of-plane manoeuvres and can easily include so-called J_2 perturbations[1].
- A new robust SLS variant is presented that, without any assumptions on the model uncertainties' structure, can guarantee satisfying constraints. Other methods can do so as well but are either too conservative or the number of required constraints scales exponentially with the number of possible sources of uncertainty.
- An overview is provided of the required solving time for varying constellation sizes for varying solvers, including a GPU solver and various toolboxes.
- An extension of the previous point is provided by finding a new, sparse formulation for the robust SLS controller that is transformed into an equivalent quadratic programme. This formulation is significantly faster than the original formulation, especially for larger problems.
- New collision avoidance constraints for satellites within the same plane and satellites in different planes are constructed.

[1]See Chapter 2 for details on what J_2 perturbations exactly are.

Chapter 2

Orbital Mechanics

To efficiently and robustly control a satellite in its orbit around the Earth, it is common to use a model-based approach. These models require an understanding of the relative physics of the satellite, which is discussed in this chapter. First, the fundamentals of orbital mechanics are discussed in Section 2-1, followed by three different models used to control the satellite in Section 2-2: the cylindrical Hill-Clohessy-Wiltshire model, the quasi-nonsingular Relative Orbital Elements model and the newly proposed Blend model. The differences between these models are demonstrated in Section 2-3. Finally, conclusions are drawn in Section 2-4

2-1 Fundamentals of Orbital Mechanics

As with almost any three-dimensional problem, a good understanding of the used reference frames is important to follow the derivations. Therefore, the most commonly used reference frames for orbital mechanics are discussed in Section 2-1-1. These are followed by the introduction of important orbital parameters in Section 2-1-2 and the usage of these parameters in absolute models in Section 2-1-3.

2-1-1 Different Frames

Three frames often used in orbital mechanics are the Earth-Centred Inertial (ECI) frame, the Local-Vertical-Local-Horizontal (LVLH) frame and the perifocal frame.

ECI Frame

The ECI frame is centred at the Earth, with the x-axis pointing along the vernal equinox, the z-axis pointing to the (geometric) north pole and the y-axis completing the right-hand coordinate system. The vernal equinox is when the Sun crosses the equator during March[1], at which point the line from the centre of the Earth to the Sun is defined as the x-axis of the ECI frame [15]. The ECI frame is visualised in Fig. 2-1a.

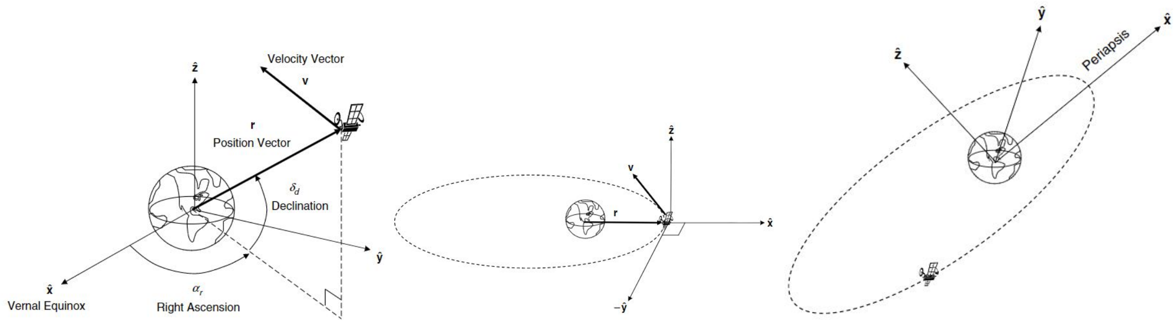
LVLH Frame

The LVLH frame is centred at the spacecraft of interest. The x-axis points away from the Earth, the z-axis is aligned with the angular momentum vector, and the y-axis completes the right-handed coordinate system. These axes are sometimes denoted as r (radial), t (tangential) or n (normal) instead of x, y and z, respectively. The LVLH frame is shown in Fig. 2-1b.

[1]For this reason, it is also called the March equinox.

Perifocal Frame

The fundamental plane of the perifocal frame is the orbital plane, where the x-axis points towards the perigee, the z-axis aligns with the angular momentum vector, and the y-axis completes the right-handed coordinate system centred at the Earth. Perigee, from the Greek words *'peri'* (near) and *'ge'* (Earth), is the closest point on the orbit to the Earth. Its counterpart is called apogee, from *'apo'* (away from), which is the point on the orbit furthest away from the Earth. The perifocal frame[2] is shown in Fig. 2-1c.



(a) ECI frame **(b)** LVLH frame **(c)** Perifocal frame

Figure 2-1: Different frames used in orbital mechanics, all from [15].

2-1-2 Orbital Parameters

As this research is focused on satellite constellations, this section discusses circular and elliptic orbits as opposed to parabolic or hyperbolic flybys. An example of a generic elliptical orbit is shown in Fig. 2-2, where several standard orbital parameters are denoted. First, several basic parameters are discussed, followed by the anomalies, orbit rotations and the introduction of orbital elements.

Basic Parameters

Two of the most critical parameters for an orbit are the semi-major axis a and the eccentricity e, both visualised in Fig. 2-2. The first, the semi-major axis a, is the most common parameter to denote the ellipse size. As the name implies, it is equal to half the length of the ellipse's major axis. The semi-major axis a plays an essential role in the angular velocity of the satellite around the Earth, as the mean angular velocity around the orbit, also called the mean motion n, can be computed with:

$$n = \sqrt{\frac{\mu}{a^3}}, \tag{2-1}$$

with μ representing the gravitational parameter of the Earth.

[2] Periapsis, as used in Fig. 2-1c, is the more general term for perigee, which is only used when the Earth is in one of the focal points of the ellipse. Its counterpart is the apoapsis, both of which stem from the Greek word '*apsis*' (arch).

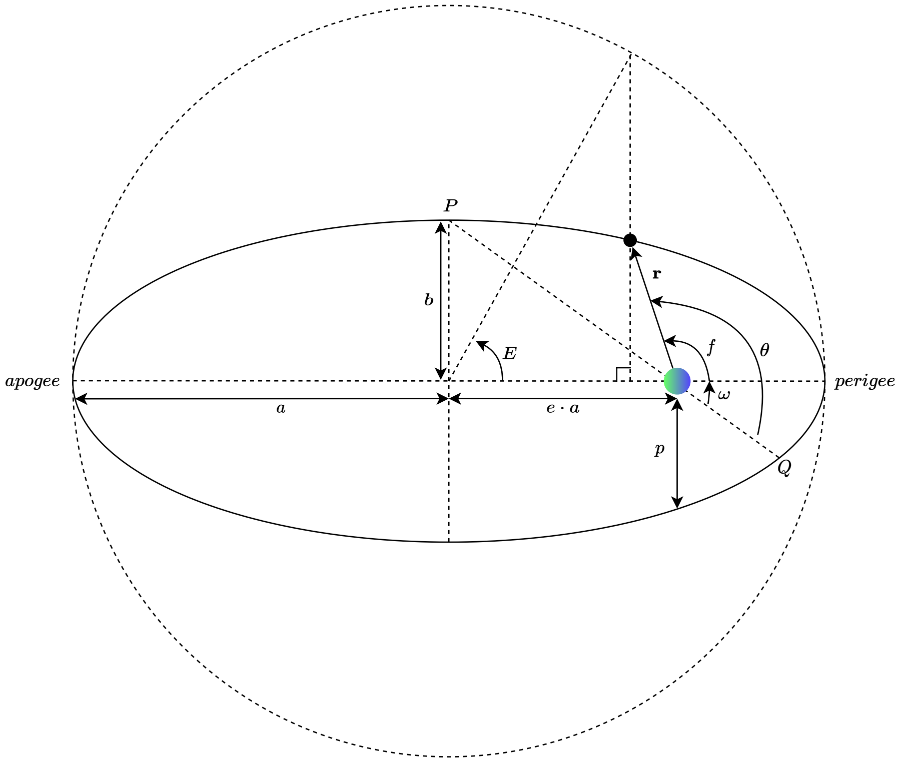


Figure 2-2: Elliptical orbit with important orbital parameters (inspired by [15]).

The second parameter, the eccentricity e, is used to represent the shape of the ellipse. For a circular orbit, the eccentricity is zero, whereas the eccentricity is between zero and one for an elliptic orbit[3]. The eccentricity relates the semi-major axis a and the semi-minor axis b, which is half the length of the minor axis of the ellipse, as:

$$b = a\sqrt{1-e^2}. \tag{2-2}$$

The last term on the right-hand side in Eq. (2-2) is often denoted as:

$$\eta = \sqrt{1-e^2},$$

to shorten the equations in the next (sub)sections.

Another distance often used in models is the semi-latus rectum p, which is the distance from the Earth towards the ellipse parallel to b. Its form is surprisingly similar to b:

$$p = a(1-e^2). \tag{2-3}$$

Finally, an important parameter not directly visible in Fig. 2-2 is the angular momentum per unit mass h, which is computed as:

$$h = \sqrt{\mu a(1-e^2)} = a^2 n\eta.$$

[3]The eccentricity can be larger or equal to one, but only in the cases of a parabolic or hyperbolic flyby.

Anomalies

The position of the spacecraft along the orbit is parameterised through one of three different anomalies, all of which describe the angle relative to the line of apsides[4]: the line connecting apogee and perigee. These three anomalies are:

- The true anomaly f is the angle between the line of apsides and the line from Earth to the spacecraft. This angle is shown in Fig. 2-2.
- The eccentric anomaly E is the angle between the line of apsides and a point on a circle with radius a, connected to the spacecraft through a line perpendicular to the line of apsides. This angle is also visualised in Fig. 2-2.
- The mean anomaly M is the angle the true anomaly f would have had if the satellite moved with a constant mean velocity given the time that has passed. The mean anomaly equals the true anomaly for a circular orbit, as the velocity is constant in that case.

The mean anomaly can be calculated as:

$$M = M_0 + n(t - t_0) = E - e \sin E, \tag{2-4}$$

with the mean motion n as in Eq. (2-1), t_0 the epoch[5] and M_0 the mean anomaly at epoch. A conversion between the mean and true anomaly is also possible, but this yields a more complex result:

$$\begin{aligned} M &= f + 2\sum_{n=1}^{\infty}(-1)^n \left(\frac{1}{n} + \sqrt{1-e^2}\right)\left(\frac{1-\sqrt{1-e^2}}{e}\right)^n \sin nf \\ &= f - 2e\sin f + \left(\frac{3}{4}e^2 + \frac{1}{8}e^4\right)\sin 2f - \frac{1}{3}e^3 \sin 3f + \frac{5}{32}e^4 \sin 4f + \mathcal{O}(e^5). \end{aligned} \tag{2-5}$$

Orbit Rotations

Given a position in a perifocal frame, the position can be expressed in the ECI frame through three Euler angles, also shown in Fig. 2-3.

- The Right Ascension of the Ascending Node (RAAN) Ω, which is the angle between the vernal equinox and the Line of Nodes (LON). For a non-equatorial orbit, there are two points (called nodes) where the satellite crosses the equatorial plane: one moving in the positive z-direction (the ascending node) and one moving in the negative z-direction (the descending node). The line connecting these two nodes is called the LON.
- The inclination angle i, which is the angle between the equatorial and orbital plane.
- The argument of periapsis ω, which is the angle between the LON and the line of apsides.

These Euler-angles chain together in the ZXZ order, where the first rotation (Ω) is around the (local) Z-axis, followed by a rotation (i) around the local X-axis and finally another rotation (ω) around the local Z-axis.

[4] Apsides is the plural form of '*apsis*'. For more information, see Footnote 2.

[5] A reference time where the states are known. A common choice is the time it passes perigee.

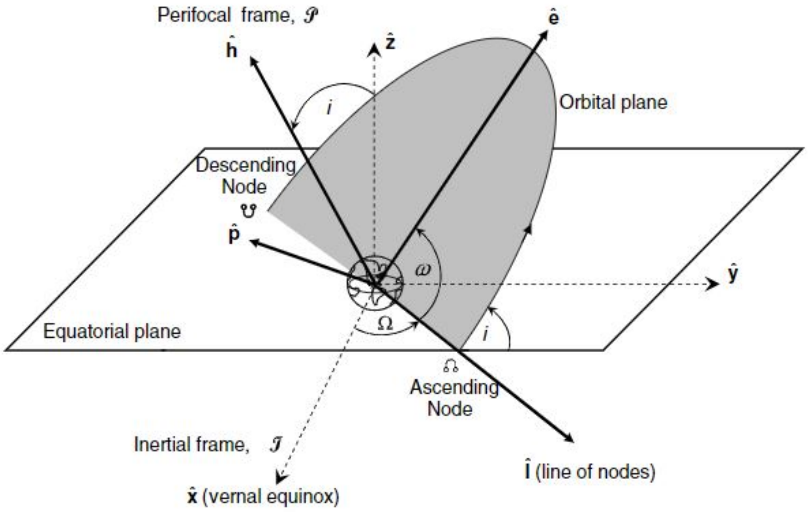


Figure 2-3: Visualisation of the Euler angles [15].

Orbital Elements

An important observation can now be made. The position of a satellite in the perifocal frame can be fully described by the semi-major axis a, the eccentricity e and an anomaly. For example, the radius of the satellite can be found using the following (nonlinear) expression:

$$r = \frac{a(1-e^2)}{1+e\cos f}. \tag{2-6}$$

Furthermore, the mapping from the perifocal frame to the ECI frame can be fully described using the Euler angles Ω, i and ω. Thus, the position of a satellite can be expressed in the ECI frame using the so-called classical orbital elements **œ**: a combination of these five parameters and an anomaly. The most common anomaly is the mean anomaly, such that the orbital elements are defined as:

$$\text{œ} = \{a, e, i, \Omega, \omega, M\},$$

where sometimes the mean anomaly at epoch M_0 is used instead, such that the position depends on the orbital elements and the time t through the equation for the mean anomaly as in Eq. (2-4).

Note, however, that these orbital elements are not well-defined for all orbits. For a circular orbit, there are no apogee or perigee, for example. The lack of a line of apsides means that the anomalies, as defined until now, lack a clear reference. A common solution is to add the argument of periapsis to these anomalies, such that the total angle is well-defined. The argument of latitude θ, which is the sum of the true anomaly f and the argument of periapsis ω, is a good example of this:

$$\theta = f + \omega. \tag{2-7}$$

Both the argument of latitude and the argument of periapsis are shown in Fig. 2-2 as well.

2-1-3 General Absolute Models

Absolute models describe the motion of an orbiting body around another, usually significantly larger body. For this thesis, these equations model the physics of a satellite in an orbit around the Earth. First, a general set of models is presented for the two-body problem, after which the effect of so-called J_2 perturbations is presented with orbital elements and Cartesian coordinates. The subsection ends with the effect of non-conservative perturbations on the orbital elements.

General Two-Body Problem

For the Keplerian two-body problem, the following assumptions are made on both bodies [15]:

- The mass of the primary body is significantly larger than that of the secondary body.
- The only force acting on the two bodies is Newtonian gravity.
- Both bodies are spherical.

These assumptions produce the following equations of motion:

$$\ddot{\mathbf{r}} + \frac{\mu}{r^3}\mathbf{r} = \mathbf{0},$$

where $r = \|\mathbf{r}\|_2$ and $\mathbf{r}$ is the position of the satellite expressed in the ECI frame. This equation often fails to provide an accurate model as the assumptions are too strict. Where the first assumption is valid for a satellite orbiting the Earth, the latter two are in practice never met. For example, drag, tidal, and third-body gravitational forces also affect the system, and as the goal is to control the satellite, a control input (and thus control force) should also be considered. Therefore, a more general formulation of the two-body problem is:

$$\ddot{\mathbf{r}} + \frac{\mu}{r^3}\mathbf{r} = \mathbf{d}, \tag{2-8}$$

where $\mathbf{d}$ denotes any perturbing force absent in the previous formulation, including a possible control input. The problem from Eq. (2-8) can be rewritten into a formulation using the orbital elements [15]:

$$\dot{\boldsymbol{\text{œ}}} = \mathfrak{P}^\top \left[\frac{\partial \mathbf{r}}{\partial \boldsymbol{\text{œ}}}\right]^\top \mathbf{d},$$

where the Poisson matrix $\mathfrak{P}^\top$ is given by:

$$\mathfrak{P}^\top = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{2}{na} \\ 0 & 0 & 0 & 0 & -\frac{\eta}{na^2 e} & \frac{\eta^2}{na^2 e} \\ 0 & 0 & 0 & -\frac{1}{na^2\eta \sin i} & \frac{\cot i}{na^2 \eta} & 0 \\ 0 & 0 & \frac{1}{na^2 \eta \sin i} & 0 & 0 & 0 \\ 0 & \frac{\eta}{na^2 e} & -\frac{\cot i}{na^2\eta} & 0 & 0 & 0 \\ -\frac{2}{na} & -\frac{\eta^2}{na^2 e} & 0 & 0 & 0 & 0 \end{bmatrix}.$$

If all forces in $\mathbf{d}$ are conservative and only position dependent with perturbing potential $\mathcal{R}$, Lagrange's Planetary Equations (LPE) are obtained:

$$\begin{aligned}\dot{\boldsymbol{œ}} &= \mathfrak{P}^\top \left[\frac{\partial \mathbf{r}}{\partial \boldsymbol{œ}}\right]^\top \mathbf{d} \\ &= \mathfrak{P}^\top \left[\frac{\partial \mathbf{r}}{\partial \boldsymbol{œ}}\right]^\top \frac{\partial \mathcal{R}}{\partial \mathbf{r}} \\ &= \mathfrak{P}^\top \frac{\partial \mathcal{R}}{\partial \boldsymbol{œ}}.\end{aligned} \tag{2-9}$$

The requirements for this potential $\mathcal{R}$ such that Eq. (2-9) holds sound strong, but they capture one of the most important disturbances in orbital mechanics: the J_2 perturbations.

J_2 Perturbations With Orbital Elements

A common potential $\mathcal{R}$ investigated in the literature is that of zonal gravitational harmonics. These harmonics account for the fact that the Earth is not a perfect sphere but an oblate spheroid with a larger radius near the equator than the poles. The varying radius of the Earth causes the gravitational field to change depending on the position in the orbit; thus, both requirements for $\mathcal{R}$ (i.e., a conservative force with position-dependent potential) are met. The potential $\mathcal{R}$ is of the form:

$$\mathcal{R} = -\frac{\mu}{r}\sum_{k=2}^{\infty} J_k(\frac{R_e}{r})^k P_k(\sin i \sin\theta), \tag{2-10}$$

where R_e is the equatorial radius of the Earth, J_k a zonal coefficient and P_k a Legendre polynomial of the first kind of order k, which one can express as:

$$P_k(x) = \frac{1}{2^k k!}\frac{d^k}{dx^k}\left[(x^2-1)^k\right].$$

For the Earth, the most dominant coefficient is the J_2 term, being almost three orders of magnitudes larger than the other terms. The first five terms are denoted in Table 2-1 to provide intuition into the magnitude of these zonal coefficients. Because the J_2 term is significantly larger than its counterparts, often only the J_2 term is considered, and Eq. (2-10) is simplified to:

$$\mathcal{R} = -\frac{\mu J_2 R_e^2}{2r^3}(3\sin^2 i \sin^2\theta - 1). \tag{2-11}$$

Table 2-1: First five zonal coefficients for the Earth.

	J_2	J_3	J_4	J_5	J_6
Zonal coefficient	$-1.08 \cdot 10^{-3}$	$2.53 \cdot 10^{-6}$	$1.61 \cdot 10^{-6}$	$2.27 \cdot 10^{-7}$	$-5.39 \cdot 10^{-7}$

When Eq. (2-11) is combined with the LPE of Eq. (2-9), several highly nonlinear equations of motions are obtained. These orbital elements with their equations of motion are called osculating orbital elements[6] and represent the exact location of the satellite at any time. However, these osculating orbital elements have two disadvantages:

1. Firstly, the osculating orbital elements result in a highly nonlinear model that is hard to use for control purposes. Without any linearisation or approximation of the equations of motion, only a select group of control algorithms can work with them.

2. Secondly, the osculating orbital elements oscillate even in a relatively stable orbit. The satellite must indefinitely provide a control input to counteract these oscillations. Even if oscillations are perfectly counteracted, they return once the control inputs are removed as the source (i.e., the oblateness of the Earth) cannot be removed.

Therefore, using mean orbital elements for control design is standard. These mean orbital elements are derived from Eq. (2-11) using the method of averaging [16]:

$$\bar{\mathcal{R}} = \frac{\bar{n}^2 J_2 R_e^2}{4(1-\bar{e}^2)^{\frac{3}{2}}}(2 - 3\sin^2\bar{i}),$$

such that, after combining this result with Eq. (2-9), the following differential equations are obtained [17]:

$$\begin{aligned}
\frac{d\,\bar{a}}{dt} &= \frac{d\,\bar{e}}{dt} = \frac{d\,\bar{i}}{dt} = 0,\\
\frac{d\,\bar{\Omega}}{dt} &= -\frac{3}{2}\frac{J_2 R_e^2 \bar{n}}{\bar{a}^2\bar{\eta}^4}\cos\bar{i},\\
\frac{d\,\bar{\omega}}{dt} &= \frac{3}{4}\frac{J_2 R_e^2 \bar{n}}{\bar{a}^2\bar{\eta}^4}(5\cos^2\bar{i} - 1),\\
\frac{d\,\bar{M}_0}{dt} &= \frac{3}{4}\frac{J_2 R_e^2 \bar{n}}{\bar{a}^2\bar{\eta}^4}\bar{\eta}(3\cos^2\bar{i} - 1),
\end{aligned}$$

where the superscript $\bar{(\cdot)}$ denotes an averaged component. For simplicity of notation, the overhead bar is dropped in future notations, and all orbital elements denote their averaged counterpart unless explicitly mentioned otherwise.

Note that the last three derivatives are constant, as these only depend on the first three (constant) derivatives. Therefore, these equations of motion are substantially easier than those for the osculating orbital elements, as they are linear for a given orbit. In general, the effect of J_2 perturbations is significant, even if the initial conditions are chosen such that the J_2 perturbations are small [18].

[6]Note that this is not a typographical error for oscillating orbital elements, but that these are, in fact, named osculating orbital elements. This comes from the Latin word '*osculare*' (to kiss), as the osculating orbital elements describe an elliptical orbit that, at that instant, 'kisses' (i.e., coincides with) the current trajectory. Due to the J_2 perturbations, the satellite does not exactly follow an elliptical orbit anymore; therefore, the equations of motion of the osculating orbital elements are highly nonlinear.

J_2 Perturbations With Cartesian Coordinates

It is important to note that Eq. (2-10) is not restricted to the use of orbital elements. The term $\sin i \sin\theta$ is equal to the sine of the geocentric latitude, and the equation can therefore be rewritten as:

$$\mathcal{R} = -\frac{\mu}{r}\sum_{k=2}^{\infty} J_k(\frac{R_e}{r})^k P_k(\frac{z}{r}),$$

where z is the z-coordinate in the ECI frame. This potential can then be combined with Eq. (2-8), which is worked out up to J_6 in [19]. For brevity, the result is here denoted with only the most significant term, J_2[7]:

$$\mathbf{d} = \begin{bmatrix} \frac{\partial \mathcal{R}}{\partial x} \\ \frac{\partial \mathcal{R}}{\partial y} \\ \frac{\partial \mathcal{R}}{\partial z} \end{bmatrix} = \frac{3\mu R_e^2 J_2}{2r^5}\begin{bmatrix} x\left(5\frac{z^2}{r^2}-1\right) \\ y\left(5\frac{z^2}{r^2}-1\right) \\ z\left(5\frac{z^2}{r^2}-3\right) \end{bmatrix}.$$

An expression for the J_2 disturbance in the LVLH frame is provided in [20], although use is made of orbital elements in their notation:

$$\mathbf{d} = \begin{bmatrix} d_r \\ d_t \\ d_n \end{bmatrix} = -\frac{3}{2}\frac{J_2\mu R_e^2}{r^4}\begin{bmatrix} 1 - 3\sin^2 i \sin^2\theta \\ \sin^2 i \sin 2\theta \\ \sin 2i \sin\theta \end{bmatrix}. \tag{2-12}$$

Non-Conservative Perturbations

Where the effect of conservative forces, such as the Keplerian gravitational force and the J_2 perturbations, has now been covered, the effect of non-conservative forces on the orbital elements has not been discussed yet. For arbitrary disturbances instead of the conservative forces in Eq. (2-9), one can use Gauss' Variational Equations (GVE), where:

$$\dot{\text{œ}} = \frac{\partial\, \text{œ}}{\partial \dot{\mathbf{r}}}\mathbf{d},$$

is transformed into:

$$\begin{aligned}
\frac{d\,a}{dt} &= \frac{2a^2 e \sin f}{h} d_r + \frac{2a^2 p}{rh} d_t, \\
\frac{d\,e}{dt} &= \frac{p \sin f}{h} d_r + \frac{(p+r)\cos f + re}{h} d_t, \\
\frac{d\,i}{dt} &= \frac{r\cos\theta}{h} d_n, \\
\frac{d\,\Omega}{dt} &= \frac{r\sin\theta}{h\sin i} d_n, \\
\frac{d\,\omega}{dt} &= -\frac{p\cos f}{he} d_r + \frac{(p+r)\sin f}{he} d_t - \frac{r\sin\theta}{h\tan i} d_n, \\
\frac{d\,M}{dt} &= \frac{b(p\cos f - 2re)}{ahe} d_r - \frac{b(p+r)\sin f}{ane(1+e\cos f)} d_t,
\end{aligned} \tag{2-13}$$

where $[d_r, d_t, d_n]$ can be any disturbance in the LVLH frame.

[7] See Table 2-1 for the first five values.

Although most non-conservative forces are relatively small and often ignored in models, a crucial non-conservative force that must be considered is the control input. Eq. (2-13) can be combined with Eqs. (2-1) to (2-3) and (2-6) to obtain the equations of motion that almost exclusively depend on the orbital elements and the control inputs:

$$
\begin{aligned}
\frac{d\,a}{dt} &= \frac{2e\sin f}{n\sqrt{1-e^2}}u_r + \frac{2(1+e\cos f)}{n\sqrt{1-e^2}}u_t, \\
\frac{d\,e}{dt} &= \frac{\sqrt{1-e^2}\sin f}{an}u_r + \frac{\sqrt{1-e^2}\,[(2+e\cos f)\cos f + e]}{an(1+e\cos f)}u_t, \\
\frac{d\,i}{dt} &= \frac{\sqrt{1-e^2}\cos\theta}{an(1+e\cos f)}u_n, \\
\frac{d\,\Omega}{dt} &= \frac{\sqrt{1-e^2}\sin\theta}{an(1+e\cos f)\sin i}u_n, \\
\frac{d\,\omega}{dt} &= -\frac{\sqrt{1-e^2}\cos f}{ane}u_r + \frac{\sqrt{1-e^2}(2+e\cos f)\sin f}{ane(1+e\cos f)}u_t - \frac{\sqrt{1-e^2}\sin\theta}{an(1+e\cos f)\tan i}u_n, \\
\frac{d\,M}{dt} &= \frac{(1-e^2)\,[(1+e\cos f)\cos f - 2e]}{ane(1+e\cos f)}u_r - \frac{(1-e^2)(2+e\cos f)\sin f}{ane(1+e\cos f)}u_t,
\end{aligned}
\tag{2-14}
$$

where $[u_r, u_t, u_n]$ denote the control inputs as accelerations in radial, tangential and normal directions.

2-2 Models

Two models are the most common in the literature for constellation control, namely the Hill-Clohessy-Wiltshire model and the quasi-nonsingular Relative Orbital Elements model. These two models are discussed in Section 2-2-1 and Section 2-2-2 respectively. A new, improved model is proposed in Section 2-2-3.

2-2-1 Hill-Clohessy-Wiltshire Model

The Hill-Clohessy-Wiltshire (HCW) model is a classic model to describe the relative motion of a satellite to its reference. The original model, developed independently by Hill [21] and Clohessy and Wiltshire [22], uses rectilinear coordinates in the LVLH frame. These equations are sometimes referred to as Hill's equations[8] or the Clohessy-Wiltshire equations, but are denoted here as the HCW equations. Their work assumes a perfectly circular orbit around a spherical Earth and that the spacecraft are close together. Especially the latter is problematic for a constellation, as the (angular) separation can be large. This led to the development of the cylindrical HCW model in [23], which is the central model of this section.

Equations of Motion

The cylindrical HCW model is derived with $[r, \varphi, z]$ being the cylindrical coordinates denoting the absolute radius, angle and height, respectively. The model uses relative coordinates, where the variables Δr, $\Delta\varphi$, and Δz describe the difference in coordinates between a chief and deputy

[8]Not to be confused with the mathematical formula for differential equations from the same author.

satellite. Here, the deputy spacecraft (denoted by the subscript $_d$) is the controlled satellite, whereas the chief spacecraft (denoted by the subscript $_c$), is the reference for the satellite. The relative variables are defined as:

$$\Delta r := r_d - r_c, \quad \Delta\varphi := \varphi_d - \varphi_c, \quad \Delta z := z_d - z_c.$$

Under the assumptions that Δr and Δz are significantly smaller than r_c and that $\Delta\dot{\varphi}$ is significantly smaller than $\dot{\varphi}_c$, the cylindrical HCW model can be obtained:

$$\begin{aligned} \Delta\ddot{r} - 3n_c^2\Delta r - 2r_c n_c \Delta\dot{\varphi} &= u_r, \\ \Delta\ddot{\varphi} + \frac{2n_c}{r_c}\Delta\dot{r} &= \frac{u_t}{r_c}, \\ \Delta\ddot{z} + n_c^2\Delta z &= u_n, \end{aligned} \tag{2-15}$$

where the derivation of this model is provided in Appendix A-1. The control inputs in radial, tangential and normal directions are denoted by u_r, u_t and u_n, respectively. This result is valid for angular differences of an arbitrary size [23], which makes the model very suitable for ring constellations where all the satellites are in a single plane.

State Space Model

Converting the model from Eq. (2-15) to a state-space representation is straightforward and yields the following result:

$$\begin{bmatrix} \Delta\dot{r} \\ \Delta\dot{\varphi} \\ \Delta\dot{z} \\ \Delta\ddot{r} \\ \Delta\ddot{\varphi} \\ \Delta\ddot{z} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n_c^2 & 0 & 0 & 0 & 2n_c r_c & 0 \\ 0 & 0 & 0 & -\frac{2n_c}{r_c} & 0 & 0 \\ 0 & 0 & -n_c^2 & 0 & 0 & 0 \end{bmatrix}}_{A_{\text{HCW}}} \begin{bmatrix} \Delta r \\ \Delta\varphi \\ \Delta z \\ \Delta\dot{r} \\ \Delta\dot{\varphi} \\ \Delta\dot{z} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \frac{1}{r_c} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{B_{\text{HCW}}} \begin{bmatrix} u_r \\ u_t \\ u_n \end{bmatrix}.$$

The result is a LTI model with six states. A_{HCW} and B_{HCW} contain ten non-zero elements out of the possible 54 entries, resulting in two relatively sparse matrices.

J_2 Perturbations

This model has proven its success in earlier works [24]. However, even for almost perfectly circular orbits, it is relatively inaccurate [25], and it is not easy to include non-keplerian forces (e.g. J_2 perturbations) in this model. For example, when looking at the J_2 perturbations in the LVLH frame in Eq. (2-12), it is possible to approximate the effect using cylindrical HCW coordinates. For example, a first-order Taylor expansion for the radial term yields:

$$\begin{aligned} \Delta d_{r,J_2} &= -\frac{3}{2}\frac{J_2\mu R_e^2}{r_d^4}(1 - 3\sin^2 i_d \sin^2\theta_d) + -\frac{3}{2}\frac{J_2\mu R_e^2}{r_c^4}(1 - 3\sin^2 i_c \sin^2\theta_c) \\ &\approx 6\frac{J_2\mu R_e^2}{r_c^5}(1 - 3\sin^2 i_c \sin^2\theta_c)\Delta r + 9\frac{J_2\mu R_e^2}{r_c^4}\sin i_c \cos i_c \sin^2\theta_c (i_d - i_c) \\ &+ 9\frac{J_2\mu R_e^2}{r_c^4}\sin^2 i_c \sin\theta_c \cos\theta_c \Delta\varphi. \end{aligned}$$

This result, however, is only valid for small deviations from the linearisation point (r_c, i_c, θ_c), meaning the model would no longer be valid for arbitrary angular differences. Furthermore, the model has no direct state for the difference in the inclination, and the model would have to be LTV to include the changing variable θ_c.

2-2-2 Quasi-Nonsingular Relative Orbital Elements Model

Orbital elements are possibly the most common method to model the dynamics of the satellite. When orbital elements are used in a relative setting, they are called Relative Orbital Elements (ROE). When using ROE, (nonlinear) combinations of the orbital elements can be used for the states. This can be done to simplify the state-space model but also because certain orbital elements are not always well-defined. This property, briefly discussed at the end of Section 2-1-2, is also the reason why there are three main types of ROE:

- The singular ROE are not uniquely defined for equatorial nor circular orbits. This occurs when an elementary set of ROE is selected, such as the individual difference between each orbital element.
- The quasi-nonsingular ROE are uniquely defined for circular orbits but not for equatorial orbits. This mostly means that one cannot simply select an anomaly or the argument of periapsis as a single state, as these are not well-defined for circular orbits[9].
- The nonsingular ROE are always uniquely defined. This can result in rather abstract terms that lose their physical interpretation.

Although it is common for satellites to be placed in circular orbits, they are often placed in orbits with an inclination of at least 30 degrees [9]. This combination of circular and non-equatorial orbits makes the quasi-nonsingular ROE an attractive solution, which is why those ROE are often encountered in the literature for these problems.

Equations of Motion

The quasi-nonsingular ROE with its state vector $\delta\boldsymbol{\alpha}_{qns}$ is defined as follows:

$$\delta\boldsymbol{\alpha}_{qns} = \begin{bmatrix} \delta a \\ \delta\lambda \\ \delta e_x \\ \delta e_y \\ \delta i_x \\ \delta i_y \end{bmatrix} = \begin{bmatrix} \frac{a_d - a_c}{a_c} \\ (M_d + \omega_d) - (M_c + \omega_c) + (\Omega_d - \Omega_c)\cos i_c \\ e_d \cos\omega_d - e_c \cos\omega_c \\ e_d \sin\omega_d - e_c \sin\omega_c \\ i_d - i_c \\ (\Omega_d - \Omega_c)\sin i_c \end{bmatrix}. \tag{2-16}$$

The equations of motion of the quasi-nonsingular ROE are defined by three different terms: the effect of Keplerian dynamics $\delta\dot{\boldsymbol{\alpha}}_{qns}^{\text{Kepler}}$, the effect of control inputs $\delta\dot{\boldsymbol{\alpha}}_{qns}^{\text{Control}}$ and the effect of J_2 perturbations $\delta\dot{\boldsymbol{\alpha}}_{qns}^{\text{J}_2}$:

$$\delta\dot{\boldsymbol{\alpha}}_{qns} = \delta\dot{\boldsymbol{\alpha}}_{qns}^{\text{Kepler}} + \delta\dot{\boldsymbol{\alpha}}_{qns}^{\text{Control}} + \delta\dot{\boldsymbol{\alpha}}_{qns}^{\text{J}_2}.$$

[9] See the end of Section 2-1-2 for why this is the case.

These terms can, using the theory from Section 2-1, all be derived as is done in Appendix A-2. This results in:

$$\delta\dot{\boldsymbol{\alpha}}_{qns}^{\text{Kepler}} = \begin{bmatrix} 0 \\ -\frac{3}{2}n_c\delta a \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \delta\dot{\boldsymbol{\alpha}}_{qns}^{\text{Control}} = \frac{1}{n_c a_c}\begin{bmatrix} 2u_t \\ -2u_r \\ -2\cos\upsilon_d\, u_t + \sin\upsilon_d\, u_r \\ -2\sin\upsilon_d\, u_t - \cos\upsilon_d\, u_r \\ \cos\upsilon_d\, u_n \\ \sin\upsilon_d\, u_n \end{bmatrix}, \tag{2-17}$$

$$\delta\dot{\boldsymbol{\alpha}}_{qns}^{\text{J}_2} = \frac{1}{8}\frac{J_2R_e^2n_c}{a_c^2\eta_c^4}\begin{bmatrix} 0 \\ -21((3\cos^2 i_c - 1)\eta_c + 5\cos^2 i_c - 1)\,\delta a - 6\sin 2i_c(3\eta_c+5)\,\delta i_x \\ -6(5\cos^2 i_c - 1)\,\delta e_y \\ 6(5\cos^2 i_c - 1)\,\delta e_x \\ 0 \\ 21\sin 2i_c\,\delta a + 12\sin^2 i_c\,\delta i_x \end{bmatrix},$$

where $\upsilon_d := M_d + \Omega_d$ is known as the mean argument of latitude. Usually, it is denoted as u but altered here to prevent confusion with the control inputs.

State Space Model

The results of Eq. (2-17) can be put into an equivalent state space model. If for simplicity the J_2 perturbations are initially ignored, the following LTV model is obtained:

$$\delta\dot{\boldsymbol{\alpha}}_{qns} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{3}{2}n_c & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A_{\text{ROE,Kep}}}\delta\boldsymbol{\alpha}_{qns} + \underbrace{\frac{1}{n_c a_c}\begin{bmatrix} 0 & 2 & 0 \\ -2 & 0 & 0 \\ \sin\upsilon_d & 2\cos\upsilon_d & 0 \\ -\cos\upsilon_d & 2\sin\upsilon_d & 0 \\ 0 & 0 & \cos\upsilon_d \\ 0 & 0 & \sin\upsilon_d \end{bmatrix}}_{B_{\text{ROE}}}\begin{bmatrix} u_r \\ u_t \\ u_n \end{bmatrix}.$$

It is also possible to put the equations of motion due to J_2 perturbations in this format, which results in the following matrix:

$$\delta\dot{\boldsymbol{\alpha}}_{qns} = \gamma\underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ A_{10} & 0 & 0 & 0 & -6\sin 2i_c(3\eta_c+5) & 0 \\ 0 & 0 & 0 & -6(5\cos^2 i_c - 1) & 0 & 0 \\ 0 & 0 & 6(5\cos^2 i_c - 1) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 21\sin 2i_c & 0 & 0 & 0 & 12\sin^2 i_c & 0 \end{bmatrix}}_{A_{\text{ROE},\text{J}_2}}\delta\boldsymbol{\alpha}_{qns},$$

with A_{10} and γ defined as:

$$A_{10} := -21((3\cos^2 i_c - 1)\eta_c + 5\cos^2 i_c - 1), \quad \gamma := \frac{1}{8}\frac{J_2R_e^2n_c}{a_c^2\eta_c^4}.$$

The combined state matrix $A_{\text{ROE,full}}$ is simply an addition of $A_{\text{ROE,Kep}}$ and $A_{\text{ROE,J}_2}$:

$$A_{\text{ROE,full}} := A_{\text{ROE,Kep}} + A_{\text{ROE,J}_2}.$$

A general advantage of the orbital elements is that they yield more accurate results by providing more extensive ranges of validity compared to the cylindrical HCW model [25]. The model only contains nine non-zero elements without the J_2 perturbations, making it slightly more sparse than the HCW model. However, when including the J_2 perturbations, the number of non-zero elements increases to fourteen, making it less sparse.

2-2-3 Blend Model

As will become more apparent in Section 2-3, the previously mentioned models have some disadvantages. More specifically:

- The cylindrical HCW model works well for constellations consisting of a single plane. As long as most of the problem is two-dimensional, the linearisation errors in the z-direction are manageable. In reality, however, constellations are always constructed from several planes, especially sizeable ones. The cylindrical HCW model uses a rectilinear coordinate (z) for the out-of-plane motion, quickly losing accuracy when the distances grow. This is the same reason why the cylindrical version is preferred over its rectilinear counterpart, as its use of a curvilinear coordinate φ makes it significantly more accurate for describing in-plane motion.

- It is hard to include the effect of J_2 perturbations into the cylindrical HCW model. Although it is possible, the model would lose two desired properties by becoming LTV and only being valid close to the linearisation point.

- The cylindrical HCW model is not based on common state variables such as the orbital elements. This makes it difficult to connect it to other results in the literature, possibly requiring more research to obtain similar results.

- The quasi-nonsingular ROE model uses the semi-major axis a to control the size of the ellipse. Although this is valid, a model with the radius r, similar to the cylindrical HCW model, is preferred. This indicates the true distance to the Earth, which is more suitable for a control objective and collision avoidance constraints.

- The quasi-nonsingular ROE is LTV, which increases the computational load when using an optimisation-based control algorithm. Where for a LTI model a substantial part of the optimisation problem of the previous iteration can be re-used, all constraints containing the system's dynamics have to be revised. Intending to scale the constellation's size, the model's effect on the computational load is vital to consider.

Equations

These shortcomings led to the development of the Blend model, a blend of ideas of the cylindrical HCW and quasi-nonsingular ROE models, where these shortcomings are all addressed. This model uses the following state variable $\delta\boldsymbol{\beta}$:

$$\delta\boldsymbol{\beta} = \begin{bmatrix} \delta r \\ \delta\lambda^f \\ \delta e_x^f \\ \delta e_y^f \\ \delta\xi_x \\ \delta\xi_y \end{bmatrix} := \begin{bmatrix} r_d - r_c \\ (f_d + \omega_d) - (f_c + \omega_c) + (\Omega_d - \Omega_c)\cos i_c \\ e_d \cos f_d - e_c \cos f_c \\ e_d \sin f_d - e_c \sin f_c \\ \cos\theta_d \tan\frac{i_d}{2} - \cos(\theta_d + \Omega_d - \Omega_c)\tan\frac{i_c}{2} \\ \sin\theta_d \tan\frac{i_d}{2} - \sin(\theta_d + \Omega_d - \Omega_c)\tan\frac{i_c}{2} \end{bmatrix}.$$

The derivative of this state without any J_2 perturbations is as follows:

$$\delta\dot{\boldsymbol{\beta}} = n_c \begin{bmatrix} r_c \delta e_y^f \\ -\frac{3}{2}\frac{1}{r_c}\delta r + \frac{1}{2}\delta e_x^f \\ -\delta e_y^f + \frac{2}{a_c n_c^2} u_t \\ \delta e_x^f + \frac{1}{a_c n_c^2} u_r \\ -\delta\xi_y + \frac{1}{a_c n_c^2 (1+\cos i_c)} u_n \\ \delta\xi_x \end{bmatrix},$$

the derivations of which are all presented in Appendix A-3. It is important to note that three assumptions have been made in these derivations:

- The eccentricity is small (i.e., the orbit is near-circular). This is a common assumption in the models used in the literature, and satellite constellations are almost exclusively designed as circular orbits.

- The error in the inclination is small. Although this sounds like a strong assumption, the different planes in a satellite constellation commonly have the same inclination.

- The error in the RAAN is small. Although not critical for a large constellation, this is the most restrictive assumption. These constellations typically consist of 30 to 70 planes that differ through their RAAN, and this work will show in Section 2-3 that this model works well with differences up to 60 degrees and can always steer a satellite to a desired orbit.

It is also possible to find a LTI model for the J_2 perturbations:

$$\begin{aligned}
\delta\dot{r}_{\mathrm{J}_2} &= \gamma r_c^{-\frac{5}{2}}(3\cos^2 i_c - 1)\delta e_y^f, \\
\delta\dot{\lambda}_{\mathrm{J}_2}^f &= -\frac{7}{2}\gamma\frac{6\cos^2 i_c - 2}{r_c^{\frac{9}{2}}}\delta r - \frac{7}{2}\gamma\frac{6\cos^2 i_c - 2}{r_c^{\frac{7}{2}}}\delta e_x^f, \\
\delta\dot{e}_{x,\mathrm{J}_2}^f &= -\gamma\frac{3\cos^2 i_c - 1}{r_c^{\frac{7}{2}}}\delta e_y^f, \\
\delta\dot{e}_{y,\mathrm{J}_2}^f &= \gamma\frac{3\cos^2 i_c - 1}{r_c^{\frac{7}{2}}}\delta e_x^f, \\
\delta\dot{\xi}_{x,\mathrm{J}_2} &= -\gamma r_c^{-\frac{7}{2}}(8\cos^2 i_c - 2)\delta\xi_y, \\
\delta\dot{\xi}_{y,\mathrm{J}_2} &= \gamma r_c^{-\frac{7}{2}}(8\cos^2 i_c - 2)\delta\xi_x,
\end{aligned}$$

with γ defined as:

$$\gamma = \frac{3}{4}J_2 R_e\sqrt{\mu}.$$

These derivations can all be found in Section A-3-5.

State-Space Model

Without any J_2 perturbations, a sparse model is obtained:

$$\begin{bmatrix}\delta\dot{r}\\ \delta\dot{\lambda}^f\\ \delta\dot{e}_x^f\\ \delta\dot{e}_y^f\\ \delta\dot{\xi}_x\\ \delta\dot{\xi}_y\end{bmatrix} = n_c\underbrace{\begin{bmatrix}0 & 0 & 0 & r_c & 0 & 0\\ -\frac{3}{2}\frac{1}{r_c} & 0 & \frac{1}{2} & 0 & 0 & 0\\ 0 & 0 & 0 & -1 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & -1\\ 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}}_{A_{\mathrm{Blend,Kep}}}\begin{bmatrix}\delta r\\ \delta\lambda^f\\ \delta e_x^f\\ \delta e_y^f\\ \delta\xi_x\\ \delta\xi_y\end{bmatrix} + \underbrace{\frac{1}{a_c n_c}\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 2 & 0\\ 1 & 0 & 0\\ 0 & 0 & \frac{1}{1+\cos i_c}\\ 0 & 0 & 0\end{bmatrix}}_{B_{\mathrm{Blend}}}\begin{bmatrix}u_r\\ u_t\\ u_n\end{bmatrix},$$

with ten non-zero elements. The J_2 perturbations yield the following state matrix:

$$\begin{bmatrix}\delta\dot{r}\\ \delta\dot{\lambda}^f\\ \delta\dot{e}_x^f\\ \delta\dot{e}_y^f\\ \delta\dot{\xi}_x\\ \delta\dot{\xi}_y\end{bmatrix} = \underbrace{\gamma r_c^{-\frac{9}{2}}\begin{bmatrix}0 & 0 & 0 & r_c^2 R & 0 & 0\\ -\frac{7}{2}S & 0 & -\frac{7}{2}r_c S & 0 & 0 & 0\\ 0 & 0 & 0 & -r_c R & 0 & 0\\ 0 & 0 & r_c R & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & -r_c T\\ 0 & 0 & 0 & 0 & r_c T & 0\end{bmatrix}}_{A_{\mathrm{Blend,J_2}}}\begin{bmatrix}\delta r\\ \delta\lambda^f\\ \delta e_x^f\\ \delta e_y^f\\ \delta\xi_x\\ \delta\xi_y\end{bmatrix},$$

where R, S and T are defined as:

$$R := 3\cos^2 i_c - 1, \quad S := 6\cos^2 i_c - 2, \quad T := 8\cos^2 i_c - 2.$$

Combined with $A_{\mathrm{Blend,Kep}}$, this results in a state matrix with seven non-zero elements.

2-3 Model Comparison

To compare these different models, they are tested in three different scenarios with a simple Model Predictive Control (MPC). The setup is explained in more detail in Section 2-3-1, after which two short individual results are shown in Section 2-3-2 as well as the effect of J_2 perturbations in Section 2-3-3. Afterwards, three scenarios are discussed. The first scenario, Section 2-3-4, is a simple planar scenario where the constellation consists of a single plane. In the second scenario, Section 2-3-5, the constellations consist of two nearby planes. Finally, in Section 2-3-6, the constellation consists of six planes with a more significant separation between the planes.

2-3-1 General Setup

Fig. 2-4 provides a simple overview of the simulation setup. A MPC controller is used to find the control inputs after which a nonlinear model finds the new orbital elements. These then need to be converted to the state variables through the equations in Section 2-2. The two main components needed to obtain these results are the controller, discussed first, and the simulation environment, which is discussed afterwards. All code for the simulation is publicly available at `https://github.com/FabianBallast/SLS_Space`.

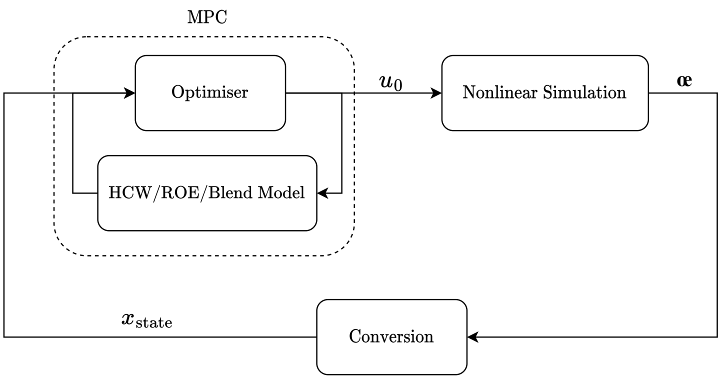


Figure 2-4: Simulation setup.

Controller Setup

To control the satellites in the simulation, a basic MPC is used. Using `Gurobi`, the following quadratic problem is solved in each iteration with a prediction horizon T of 20 steps:

$$
\begin{aligned}
\underset{\substack{x_0,\ldots,x_T,\\ u_0,\ldots,u_{T-1}}}{\text{minimize}} \quad & \sum_{t=0}^{T-1} \left[(x_t - x_r)^\mathsf{T} Q\, (x_t - x_r) + u_t^\mathsf{T} R\, u_t \right] + (x_T - x_r)^\mathsf{T} Q_T\, (x_T - x_r) \\
\text{subject to} \quad & x_{t+1} = A_t\, x_t + B_t\, u_t, \quad \forall t \in \mathbb{Z}_0^{T-1}, \\
& x_{\min} \leq x_t \leq x_{\max}, \quad \forall t \in \mathbb{Z}_1^{T}, \\
& u_{\min} \leq u_t \leq u_{\max}, \quad \forall t \in \mathbb{Z}_0^{T-1},
\end{aligned}
$$

where $\mathbb{Z}_i^j$ is set of integer numbers from i up to and including j, e.g. $\mathbb{Z}_0^{T-1} = \{0, \ldots, T-1\}$.

The exact control parameters for each model are denoted in Appendix B-1, where the parameters are selected to obtain similar performance for all models. The most important state constraint is added to the radius, which should not exceed 0.1 meters from its reference.

Simulation Setup

A vital part of the simulation is the integration of the nonlinear dynamics that represent the true dynamics. The fourth-order Runge-Kutta (RK4) method is applied to simulate the mean orbital elements, the details of which are explained further in Appendix C-1. An important parameter is Δt_{sim}, which denotes the integration time step.

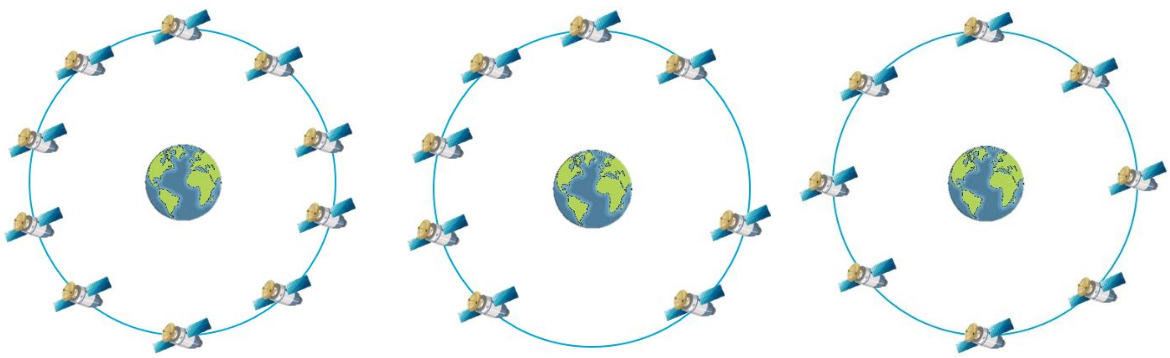
Although the simulation would ideally be performed with the true values for constants such as the gravitational parameter (μ), these parameters are changed for two reasons:

- Firstly and most importantly, the vast range in the size of the variables makes solving the optimisation problem much harder for computational reasons. For example, the gravitational parameter μ is equal to $3.986 \cdot 10^{14}$ $\text{m}^3\,\text{s}^{-2}$. Combining this with an orbital radius of 8000 km leads to the derivative of $\dot{r}$ often being large, whereas the effect of control inputs on $\dot{\theta}$ is small.

- Secondly, by changing the parameters it is also possible to speed up the simulation, as it is possible to increase the mean motion. For the HCW model the maximum relative angular velocity should be (significantly) smaller than the mean motion. Therefore, an increased mean motion allows for a larger relative angular velocity.

Note that this is not uncommon, as the same approach is followed in [24]. The same parameters as in [24] are used, denoted in Table 2-2 among other parameters. Parameters that change for each scenario, such as the number of satellites, are provided at each scenario individually.

Initial positions and reference positions are selected simplistically, with most state variables being zero except for those referring to the (relative) argument of latitude. The scenario focuses on a constellation where the satellites must reconfigure themselves to be evenly distributed. The details of the approach to select the initial and reference positions are provided in Appendix C-2, and the concept of this scenario is visualised in Fig. 2-5.



(a) Before simulation. **(b)** Start of simulation. **(c)** End of simulation.

Figure 2-5: Overview of the scenario.

Table 2-2: Parameters during the simulation.

Symbol	Meaning	Value
μ	Gravitational parameter of the Earth	$100\ \mathrm{m^3\,s^{-2}}$
R_e	Equatorial radius of the Earth	50 m
r_c	Desired orbital radius	55 m
Δt_{sim}	Simulation integration time step	1 s
Δt_{con}	Controller sampling time	10 s
T	Prediction horizon	20 steps
m	Mass of the satellite	400 kg

Metrics

The different models are compared through three primary states: the relative radius δr, the relative argument of latitude $\delta\theta$ and the relative RAAN $\delta\Omega$. These states are selected because they best describe the trajectory of the satellites and represent clear physical values. To compare the results numerically, the following metric is used to quantify the control inputs for a simulation of duration T_{sim}:

$$\|\bar{\mathbf{u}}\|_2 := \frac{1}{T_{\text{sim}}} \int_0^{T_{\text{sim}}} \sqrt{u_n(t)^2 + u_t(t)^2 + u_z(t)^2}\, dt,$$

where, in practice, the discrete alternatives are used to compute the values. Furthermore, the average computation time it takes to find the optimal control input for each model is denoted by T_{comp}. For a relative comparison between the different models, a normalised variant of these two metrics is provided as well, denoted by $\|\bar{\mathbf{u}}\|_2^{\text{norm}}$ and $T_{\text{comp}}^{\text{norm}}$, respectively. These normalised metrics are divided by the smallest value across all models.

This space was intentionally left blank.

2-3-2 Individual Results

This subsection contains two results for the HCW and Blend models. The projection issues of the HCW model are discussed first, followed by finding the maximum out-of-plane distance for the Blend model.

Projection Problems With The Hill-Clohessy-Wiltshire Model

The cylindrical HCW model places the origin of its coordinate system for a satellite at its corresponding reference satellite. The in-plane coordinates of the controlled satellite are then projected onto the orbital plane of the reference satellite, which can cause a distorted representation of the orbit. To illustrate this, a simulation is run without a controller and with four different RAANs. The simulation details can be found in Table 2-3.

Table 2-3: Parameters during the projection issue scenario.

Symbol	Meaning	Value
a	Semi-major axis	55 m
e	Eccentricity	0
i	Inclination	45 deg
ω	Argument of periapsis	0 deg
Ω	RAAN	$\{0, 15, 30, 60\}$ deg
N_{sat}	Number of satellites in the simulation	4 satellites

The first plot in Fig. 2-6 shows these orbits in a three-dimensional view, where all orbits are transformed such that the orbit with an Ω of 0 degrees lies in the xy-plane. The second plot shows the distorted orbit for the in-plane states. Where the difference is small for planes with a similar RAAN, the circular orbit becomes elliptical for a RAAN of 60 degrees. The effect on the out-of-plane state z is visible for all orbits with separations of several meters.

The corresponding relative radial and height states are shown in Fig. 2-7. Note how the orbit with a Ω of 15 degrees has a maximum radial error of 1 m, which is ten times the maximum allowed deviation in the simulations. This plot shows that the distorted orbits can fail to meet the constraints when the actual orbit satisfies all constraints. This is an inherent limitation of the cylindrical HCW model with out-of-plane manoeuvres.

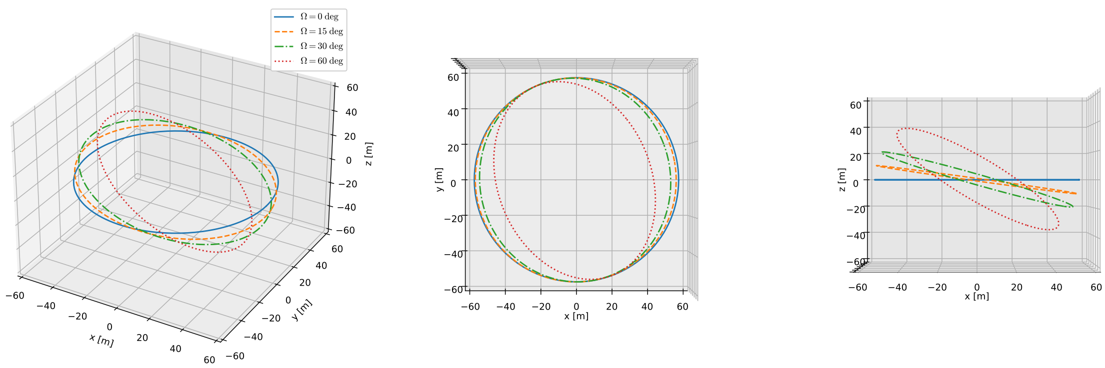


Figure 2-6: 3D-view of projected orbits with Hill-Clohessy-Wiltshire model.

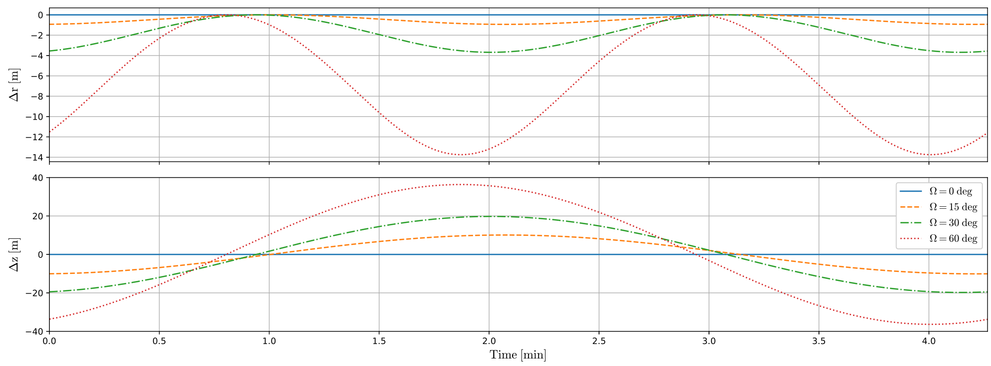


Figure 2-7: States of projected orbits with Hill-Clohessy-Wiltshire model.

Maximum Out-Of-Plane Distance With The Blend Model

In Section 2-2-3 an assumption is made that the distance between planes in terms of the RAAN is not too large. To quantify this limit, it is tested to what extent this distance can be increased before the performance degrades substantially. The parameters of this scenario are shown in Table 2-4, where all satellites start with a reference RAAN of 180 degrees.

Table 2-4: Parameters during the maximum out-of-plane distance scenario.

Symbol	Meaning	Value
a	Semi-major axis	55 m
e	Eccentricity	0
i	Inclination	45 deg
ω	Argument of periapsis	0 deg
Ω	RAAN	$\{0, 15, 30, \ldots, 330, 345\}$ deg
N_{sat}	Number of satellites in the simulation	24 satellites

The results with the Blend model for planar distances up to 60 degrees are shown in Fig. 2-8. This includes distances of 0, 15, 30, 45 and 60 degrees in Ω in both directions. The Blend model steers all satellites to their reference, although for larger distances, the differences between satellites with a similar initial distance but different directions grow.

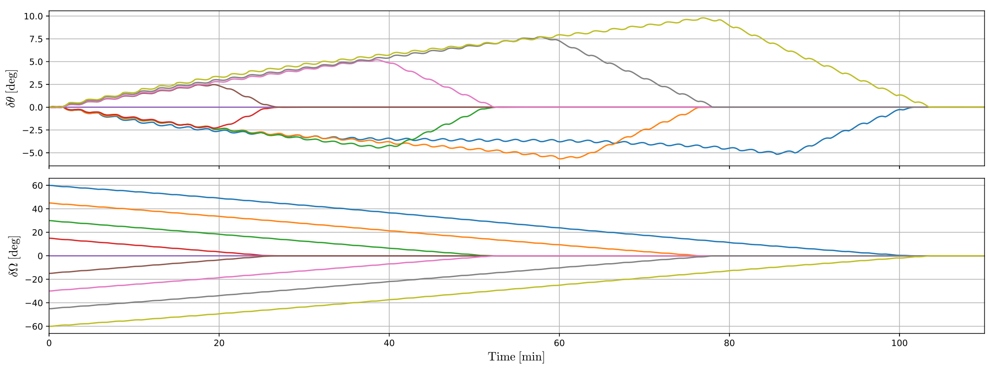


Figure 2-8: Results for Blend model for planar distances up to 60 degrees.

When looking at the satellites that had a large initial distance, it can be seen in Fig. 2-9 that the Blend model can steer all satellites to their reference. However, it is clear that, for larger distances, the followed path is substantially different than for smaller distances and is likely suboptimal.

In the extreme case of an initial error of 180 degrees, the distance to the reference initially grows as the errors for both the radius and argument of latitude grow while the error for the RAAN stays almost constant. The simulation takes almost two hours before the error is reduced, which is done rapidly after that point. For optimal performance with the Blend model, the maximum planar distance is therefore limited to approximately 60 degrees.

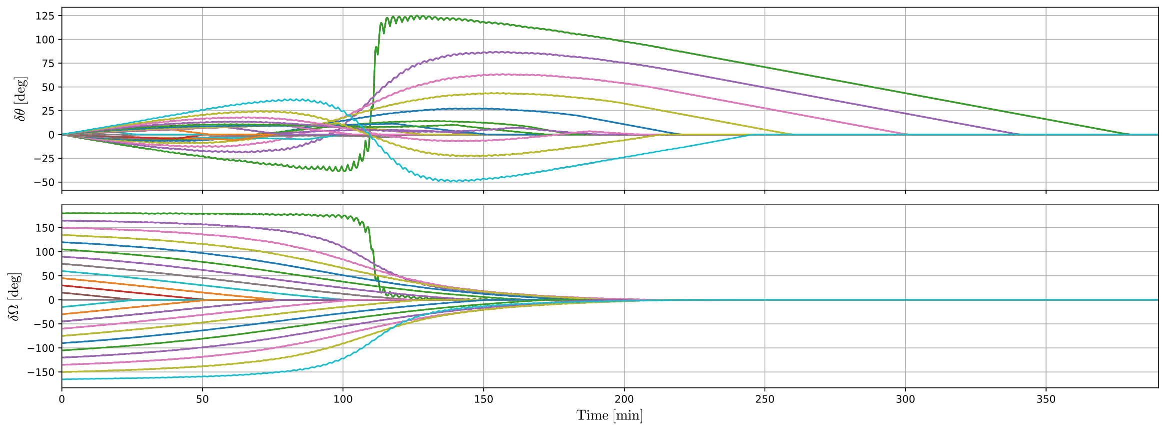


Figure 2-9: Results for Blend model for planar distances up to 180 degrees.

2-3-3 J_2 Perturbations

To show the effect of the J_2 perturbations on the model and the dynamics, a simulation is run without any control inputs. The different models, however, still try to predict the error over time. This is done for five different models:

- The `HCW` model is the model discussed in Section 2-2-1.
- The `ROE(J2)` model is the quasi-nonsingular ROE model of Section 2-2-2 with the J_2 perturbations taken into account. This is in contrast to the `ROE(NO J2)` model, where the matrix $A_{\text{ROE},\text{J}_2}$ is ignored.
- A similar approach is followed for the Blend model, where the `BLEND(J2)` model is the complete Blend model as discussed in Section 2-2-3. The `BLEND(NO J2)` model is the Blend model without the $A_{\text{Blend},\text{J2}}$ matrix.

The results for these models with an initial radial offset of 0.1 m from the reference are shown in Fig. 2-10, where the errors are computed by subtracting the predicted model states from the true states computed by the nonlinear simulation. The following conclusions can be drawn from this result:

- The HCW model is the least accurate and produces mainly for the angular state $\delta\theta$ significant errors.

- For both the ROE and Blend model are more accurate predictions achieved for the angular state $\delta\theta$ when the J_2 perturbations are included in the model. Where the ROE model can also improve the accuracy for the out-of-plane state $\delta\Omega$, the Blend model falls short in this aspect. This is as expected, as the model for the out-of-plane states of the Blend model, ξ_x and ξ_y, does not depend on the radial error. The model would have become nonlinear otherwise.

- The effect of the J_2 perturbations is generally small, as this simulation lasts six hours and the maximum error is 1.4 degrees. However, for satellites with very limited propulsion available, accurate long-term predictions can be crucial.

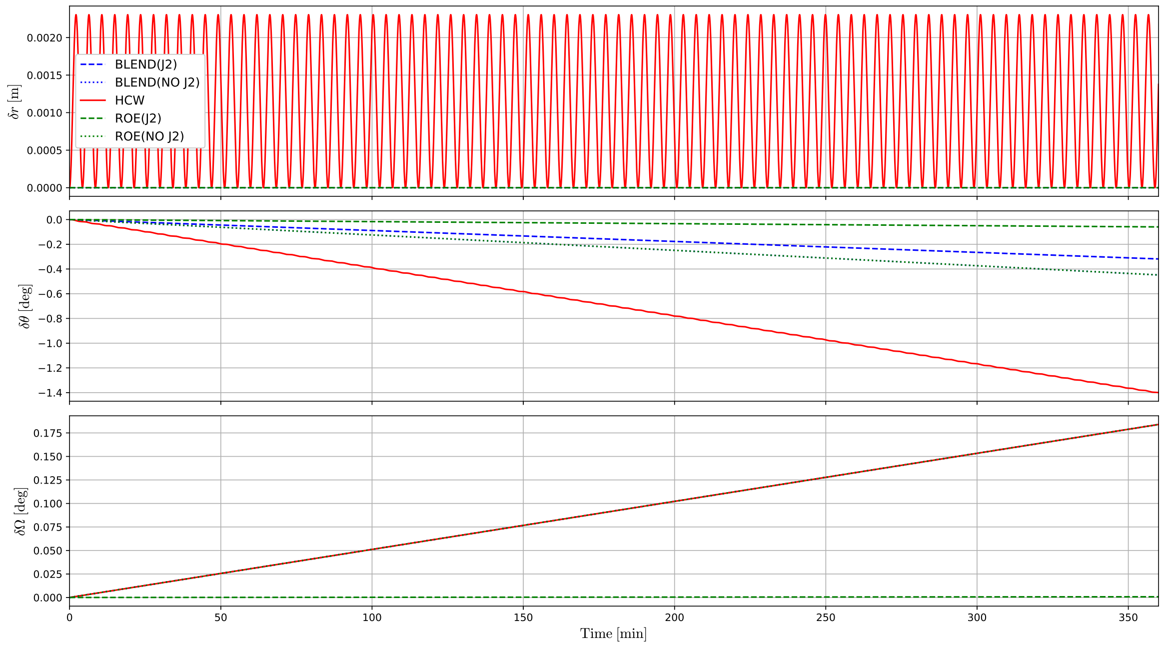


Figure 2-10: Effect of including J_2 perturbations in the model.

2-3-4 Planar Scenario

The planar scenario has all the satellites within one plane as defined by the parameters in Table 2-5. To see the full simulation results for all satellites and all controllers, see Appendix F.

Table 2-5: Parameters during the planar scenario.

Symbol	Meaning	Value
a	Semi-major axis	55 m
e	Eccentricity	0
i	Inclination	45 deg
ω	Argument of periapsis	10 deg
Ω	RAAN	20 deg
N_{sat}	Number of satellites in the simulation	10 satellites

The results for this scenario are plotted in Fig. 2-11 where only three of the ten satellites are plotted to prevent visual cluttering. Where the HCW and Blend model produce almost indistinguishable results, the quasi-nonsingular ROE shows a significant difference for δr and $\delta\theta$.

This is expected as the HCW and Blend model are closely related (see the end of Section A-3-3) and because the quasi-nonsingular ROE is the only model of these three that contains the semi-major axis a as a state variable instead of the radius r. This results in a controller that makes sub-optimal use of the allowed radius. The ROE model can perform similarly in $\delta\theta$ to the HCW and Blend model, but this requires a larger radius than allowed.

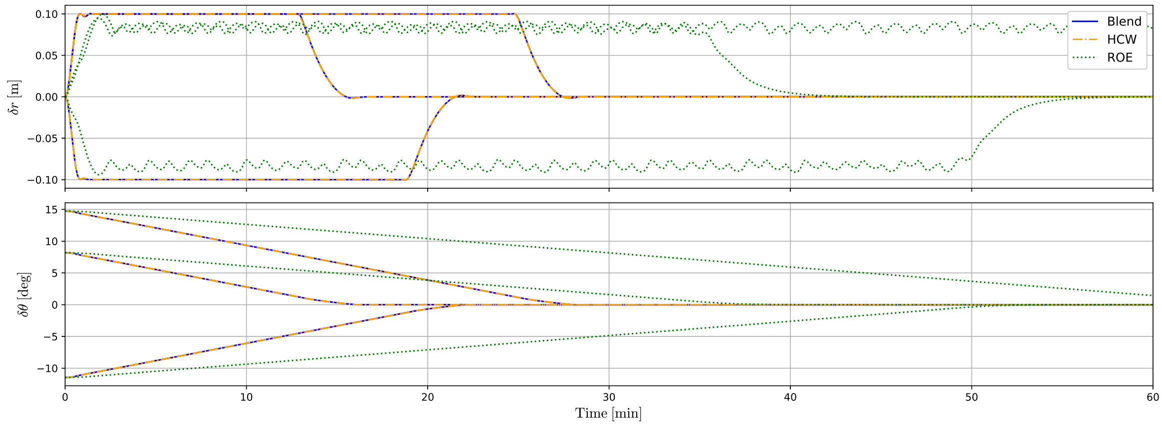


Figure 2-11: Comparison of planar states of three models in a single plane.

The corresponding control inputs for these three satellites are shown in Fig. 2-12. The radial input shows similar behaviour for the HCW and Blend model with almost constant values before dropping to zero. The ROE model is more conservative but also shows oscillatory behaviour. The tangential input oscillates quite heavily with the Blend model, whereas this is significantly less for the HCW and ROE models.

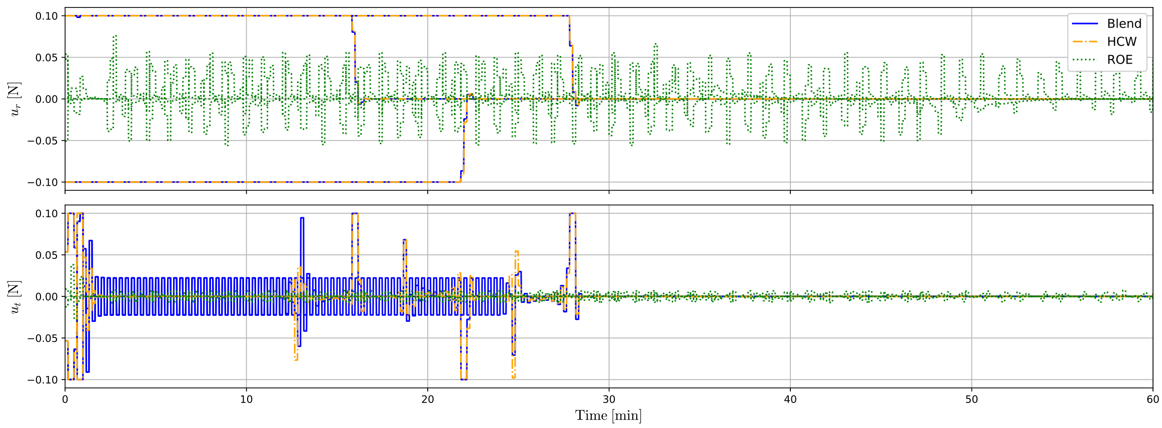


Figure 2-12: Comparison of the inputs of three models in a single plane.

These observations for the inputs are backed by the metrics as shown in Table 2-6. Although the ROE model moves the satellites slowly towards their reference, it uses considerably less input than the HCW and Blend models. Not visible in the plots is the solver time T_{comp}, where the LTI models (the Blend and HCW models) outperform the LTV ROE model by a significant margin, as these LTI models are close to four times faster than the ROE model.

Table 2-6: Metrics for single plane simulation.

Model	$\lVert\bar{\mathbf{u}}\rVert_2$ [N]	$\lVert\bar{\mathbf{u}}\rVert_2^{\text{norm}}$ [-]	T_{sol} [s]	$T_{\text{sol}}^{\text{norm}}$ [-]
Blend	0.0282	3.4443	0.0275	1.0208
HCW	0.0279	3.4124	0.0269	1.0000
ROE	0.0082	1.0000	0.1037	3.8516

2-3-5 Small Inter-Planar Scenario

In the small inter-planar scenario, two similar planes differ only in their RAAN. The exact values of these orbits are shown in Table 2-7, where the values for the RAAN are relatively close together.

Table 2-7: Parameters during the small inter-planar scenario.

Symbol	Meaning	Value
a	Semi-major axis	55 m
e	Eccentricity	0
i	Inclination	45 deg
ω	Argument of periapsis	10 deg
Ω	RAAN	$\{0, 5\}$ deg
N_{sat}	Number of satellites in the simulation	10 satellites

When taking a look at the primary states in Fig. 2-13, it is interesting to see the similarity in performance for all models in the out-of-plane state $\delta\Omega$, even though all three models use entirely different state variables to model these dynamics.

Something that might be missed at first glance is the oscillation for the HCW model in the δr state around four to five minutes into the simulation. Where previously the HCW model had continuously tracked the Blend model almost perfectly, it shows a significant deviation here. This also leads to a violation of the constraint for the radial limit, which is set to 0.1 m.

This violation is caused by the fact that the HCW model projects the current orbit onto the plane of the reference orbit. If these orbits are the same, as in the previous subsection, there is no error for the in-plane states. However, now that these orbits are different, the projected radius used in the HCW model is no longer equal to the exact radius of the orbit. This phenomenon is discussed in more detail in Section 2-3-2.

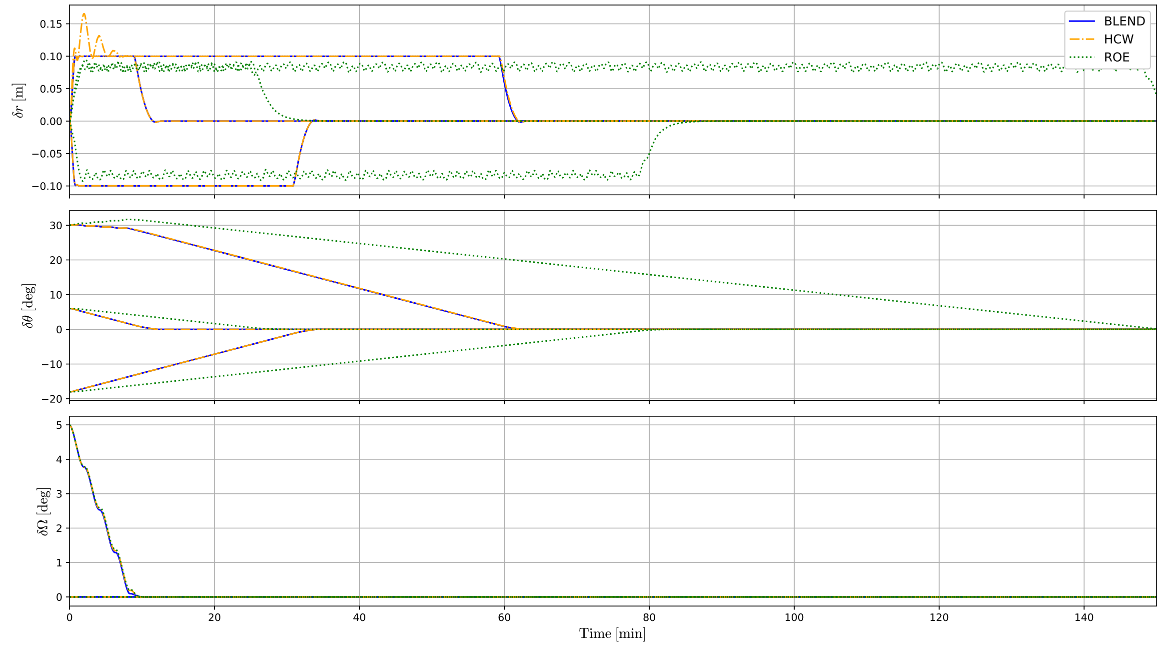


Figure 2-13: Comparison of main states of three models in two nearby planes.

For most of the simulation, all models have a similar value for u_n as shown in Fig. 2-14. The most considerable difference occurs here just before the satellite reaches the desired orbit, where the Blend model uses the least control input.

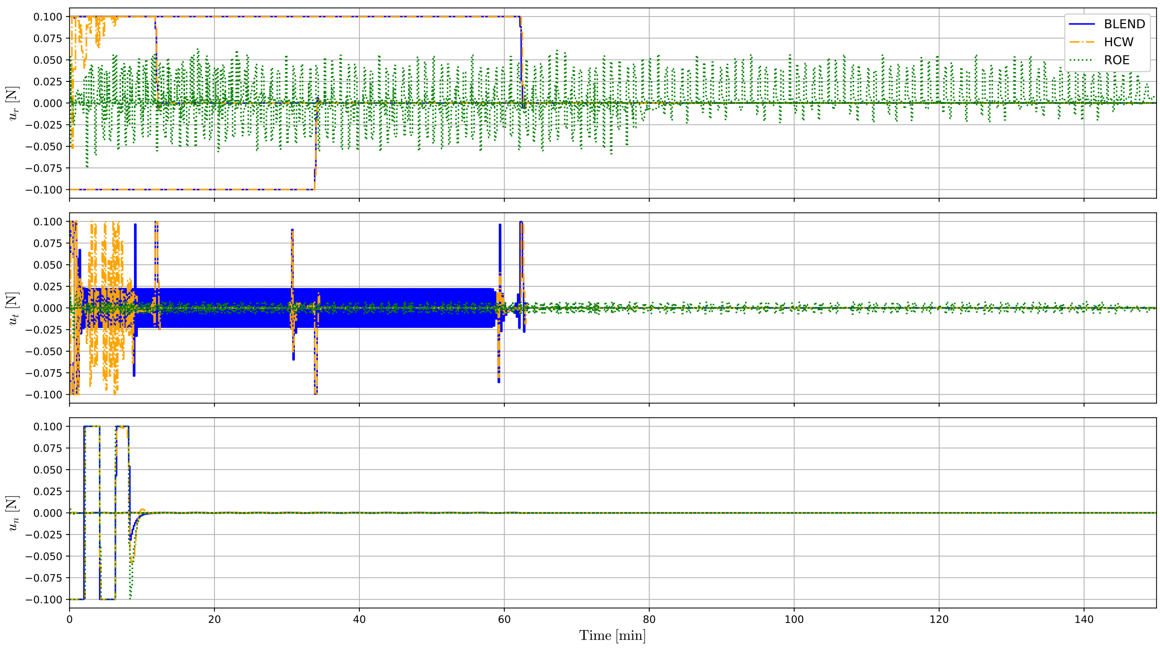


Figure 2-14: Comparison of the inputs of three models in two nearby planes.

The metrics, as shown in Table 2-8, show the generally familiar results. The HCW and Blend model have larger control inputs and require less than a fifth of the time to solve the optimisation problems for the MPC compared to the quasi-nonsingular ROE.

Table 2-8: Metrics for two nearby planes.

Model	$\|\bar{\mathbf{u}}\|_2$ [N]	$\|\bar{\mathbf{u}}\|_2^{\text{norm}}$ [-]	T_{sol} [s]	$T_{\text{sol}}^{\text{norm}}$ [-]
Blend	0.0270	3.1864	0.0280	1.0231
HCW	0.0267	3.1548	0.0274	1.0000
ROE	0.0085	1.0000	0.1380	5.0401

2-3-6 Large Inter-Planar Scenario

Previously, the projection issues that arise when using the HCW model in multi-planar scenarios have been discussed. For the following scenario, the distance between the planes is increased to 15 degrees, at which the HCW model is no longer usable. Section 2-3-2 shows that the projection errors are more than a meter in the radial direction, which is ten times the allowed limit.

The quasi-nonsingular ROE and the Blend model can deal with these scenarios. The parameters for this scenario are provided in Table 2-9.

Table 2-9: Parameters during the large inter-planar scenario.

Symbol	Meaning	Value
a	Semi-major axis	55 m
e	Eccentricity	0
i	Inclination	45 deg
ω	Argument of periapsis	10 deg
Ω	RAAN	$\{0, 15, 30, 45, 60, 75\}$ deg
N_{sat}	Number of satellites in the simulation	36 satellites

This space was intentionally left blank.

The simulation results for the two models are shown in Figs. 2-15 and 2-16. The differences between the ROE and Blend model are again visible in the radial state and the radial control input.

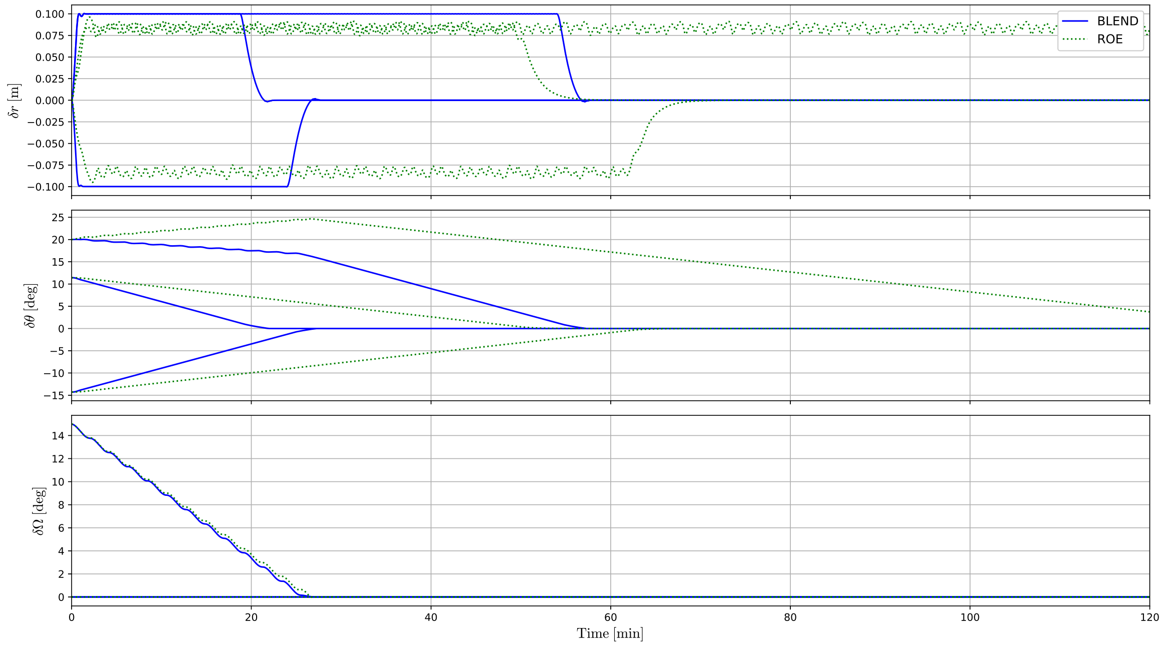


Figure 2-15: Comparison of main states of three models in six planes.

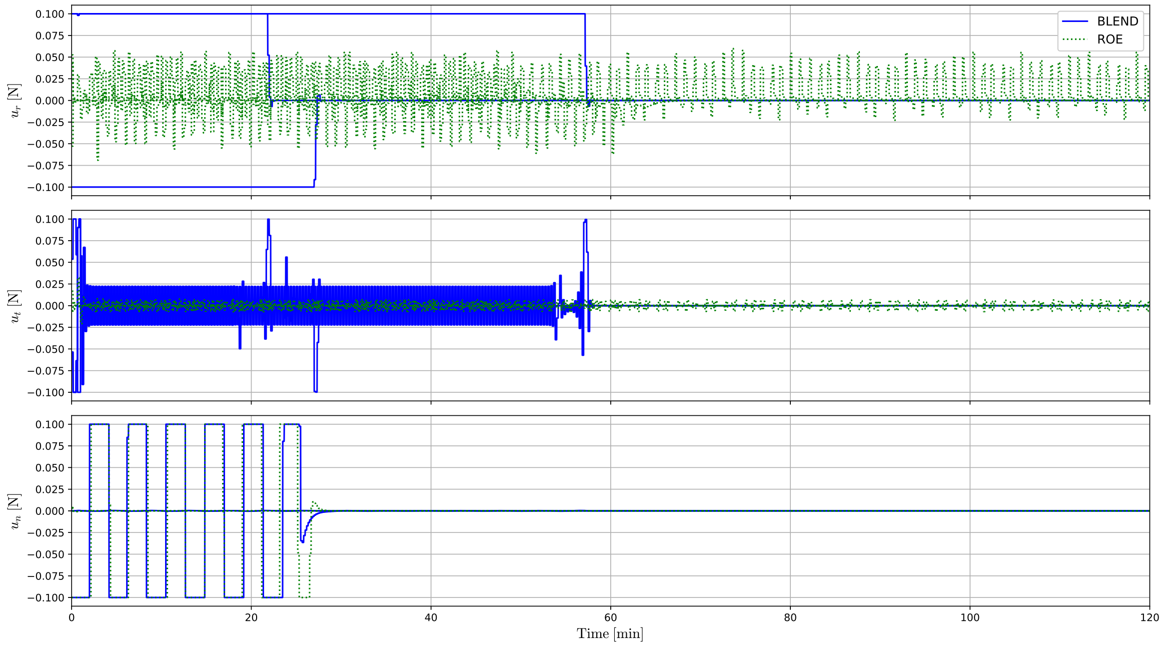


Figure 2-16: Comparison of the inputs of three models in six planes.

Finally, the metrics match the aforementioned differences as shown in Table 2-10. For this larger problem, the ROE model remains three times as effective with the control inputs. Furthermore, although the ROE model still has a larger computation time, the relative ratio has decreased compared to the previous scenarios from 4 and 5 to 2.5.

Table 2-10: Metrics for six planes.

Model	$\|\bar{\mathbf{u}}\|_2$ [N]	$\|\bar{\mathbf{u}}\|_2^{\text{norm}}$ [-]	T_{sol} [s]	$T_{\text{sol}}^{\text{norm}}$ [-]
Blend	0.0385	3.3105	0.1350	1.0000
ROE	0.0116	1.0000	0.3276	2.4272

2-4 Conclusion

Three models have been discussed in this chapter: the cylindrical HCW model, the quasi-nonsingular ROE model and the newly proposed Blend model. Where the HCW model works well for in-plane trajectories, its requirement of a projection onto the reference plane decreases the performance rapidly for out-of-plane trajectories. The ROE model works well for out-of-plane movements but is computationally more demanding as it is LTV, and it is hard to control the orbital radius. Despite the Blend model using more significant control inputs than the ROE model, it is found to be the best model for these constellations because:

- It can steer satellites with any planar separation to their references, although it works most effectively for planar separations up to 60 degrees.
- It is LTI.
- It is possible to include J_2 perturbations into the model.
- It is based upon the orbital elements, which allows it to be extended using other results in the literature.
- It can directly control the orbital radius.

To maximise the advantages of this model, it is important to use it in an appropriate model-based control algorithm. The selection of a control algorithm is discussed next in Chapter 3.

Chapter 3

Robust Control

The space industry has high standards regarding the performance of their missions. Spacecrafts should stay within their allocated orbits at all times despite any unforeseen disturbances or modelling errors. Collisions with other orbiting bodies, such as other spacecraft, space debris or meteorites, should be avoided at all costs due to the high costs and because the resulting space debris makes future space missions harder. Therefore, a robust control method guaranteed to stay within its allowed space is vital for a space mission.

This chapter discusses several available control methods in Section 3-1, where it is argued that a robust System-Level Synthesis approach is best suited here. The basics of System-Level Synthesis are then discussed in Section 3-2, followed by the robust variant in Section 3-3. Finally, the chapter closes with the results when using this method in Section 3-4 and the conclusion in Section 3-5.

3-1 Robust Control Methods

There is a plethora of available control methods available that provide some robustness. To make an initial selection, MPC is chosen as the basis of the controller. This is done for the following reasons:

- It is intuitive to balance different objectives, such as a trade-off between fuel consumption and tracking of the target.
- Although a model of the plant is required, modelling the orbital dynamics is no problem and linear models have been used for satellites before as discussed in Section 2-2. These models can be used to fully exploit the dynamics of the plant, which decreases the conservatism in the robust controller.
- It is possible to include constraints into the problem, which is not possible with, for example, (standard) Linear-Quadratic Regulator (LQR).
- MPC can easily handle multi-variable problems with coupled states. This is desired as the in-plane states in the HCW and Blend model in Section 2-2 are coupled.
- MPC inherently provides some robustness [26].

However, a large disadvantage of MPC is its computational load, where it is required to solve an optimisation problem online each time step. Luckily, the models described in Section 2-2 are linear, and it is shown in Section 2-3 that a basic MPC can control satellites with these models. This means that using a nonlinear MPC variant is unnecessary, and solving a quadratic problem suffices. The next chapter, Chapter 4, shows that this problem can be scaled to hundreds or thousands of satellites and solved within a second.

Robust Model Predictive Control Methods

There are three robust MPC methods that are discussed in more detail. The first is the tube MPC of [26]. This is the more classical approach to robust MPC, and it is also used as a benchmark in the two papers describing the other two methods [27], [28]. With tube MPC, a control law and a tube are jointly optimised such that the control law is guaranteed to keep the system within the tube despite disturbances or model errors.

The second approach is the net-additive uncertainty approach of [27]. Here, the parametric uncertainty and additive disturbance are lumped together into a single augmented disturbance. The maximum augmented disturbance is calculated first, after which the system is controlled with this maximum disturbance considered. Although this can be very restrictive, it can have a similar region of attraction as the tube MPC while being computationally significantly faster, especially for larger systems or for longer prediction horizons [27].

The final method is the lumped SLS method of [28]. This is the most novel method and builds upon the relatively new SLS framework. Here, instead of directly optimising the states and control inputs, the closed-loop transfer functions for the states and control inputs are optimised. One of the advantages of this approach is its application of robust control with scalable and non-conservative problem formulations. The lumped SLS method uses similar lumped disturbances as the net-additive uncertainty approach of [27] but uses time-varying disturbances based on the dynamics of the plant to decrease the conservativeness. The lumped SLS method is compared to both the tube MPC and net-additive uncertainty MPC in [28], where it outperforms both methods in terms of conservativeness and outperforms the tube MPC with its computational speed. Due to the time-varying lumped disturbance, it is slower than the net-additive uncertainty MPC, but this difference is relatively small.

As the lumped SLS method generally outperforms the other two methods, the method is discussed in further detail in the following two sections. First, the basics of SLS are discussed in Section 3-2, after which the lumped extension is covered in Section 3-3.

3-2 Nominal System-Level Synthesis

System-Level Synthesis (SLS) is a new framework that finds an optimal control input by optimising over a closed-loop transfer function. It was originally developed to scale gracefully for distributed systems with sparsity constraints [29], but has also been used with data-driven [30], [31] and robust [28], [32], [33] variations.

The SLS theory includes results for both state- and output-feedback, as well as theory for a finite and infinite horizon. As in practice, as well as for the lumped SLS, the finite horizon state-feedback variant is the most common; this is presented next.

First, the notations and basics of SLS are presented in Section 3-2-1. Next, constraints are added to the SLS problem in Section 3-2-2 and the complete optimisation problem is presented in Section 3-2-3.

3-2-1 Notations

For the SLS, the following LTV system is considered:

$$x_{t+1} = A_t\, x_t + B_{1,t}\, w_t + B_{2,t}\, u_t, \tag{3-1}$$

with state vector x_t, control input u_t and disturbance w_t at time t. The evolution of these variables is described through the signals $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{w}$:

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_T \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_T \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} x_0 \\ w_0 \\ w_1 \\ \vdots \\ w_{T-1} \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} K^{0,0} & & & \\ K^{1,1} & K^{1,0} & & \\ \vdots & \ddots & \ddots & \\ K^{T,T} & \cdots & K^{T,1} & K^{T,0} \end{bmatrix},$$

along with a causal LTV state-feedback controller $\mathbf{K}$, such that $\mathbf{u} = \mathbf{K}\mathbf{x}$. Note that it is standard in the SLS framework to put the initial state, x_0, as the first entry of $\mathbf{w}$. Then, if one defines:

$$\mathcal{A} := \begin{bmatrix} A_0 & & & \\ & \ddots & & \\ & & A_{T-1} & \\ & & & 0 \end{bmatrix}, \quad \mathcal{B}_2 := \begin{bmatrix} B_{2,0} & & & \\ & \ddots & & \\ & & B_{2,T-1} & \\ & & & 0 \end{bmatrix}, \tag{3-2}$$

and the disturbance acting on the state $\boldsymbol{\delta}_x$:

$$\boldsymbol{\delta}_x := \mathcal{B}_1 \mathbf{w} = \begin{bmatrix} I & & & \\ & B_{1,0} & & \\ & & \ddots & \\ & & & B_{1,T-1} \end{bmatrix} \mathbf{w} = \begin{bmatrix} x_0 \\ B_{1,0}\, w_0 \\ \vdots \\ B_{1,T-1}\, w_{T-1} \end{bmatrix},$$

the relationship between the signals according to Eq. (3-1) can be expressed as:

$$\begin{aligned} \mathbf{x} &= Z\mathcal{A}\,\mathbf{x} + Z\mathcal{B}_2\,\mathbf{u} + \boldsymbol{\delta}_x \\ &= Z(\mathcal{A} + \mathcal{B}_2\mathbf{K})\mathbf{x} + \boldsymbol{\delta}_x. \end{aligned}$$

where Z is the block-downshift operator, a matrix with identity matrices along its first block subdiagonal. Note that if the system is LTI, the matrices $\mathcal{A}$, $\mathcal{B}_1$ and $\mathcal{B}_2$ repeat the same matrix along the diagonal (except for the first entry of $\mathcal{B}_1$).

Using the feedback law $\mathbf{u} = \mathbf{K}\mathbf{x}$, the closed-loop maps from $\boldsymbol{\delta}_x$ to $\mathbf{x}$ and $\mathbf{u}$ can be described by:

$$\begin{aligned} \mathbf{x} &= (I - Z(\mathcal{A} + \mathcal{B}_2\mathbf{K}))^{-1}\boldsymbol{\delta}_x \\ \mathbf{u} &= \mathbf{K}(I - Z(\mathcal{A} + \mathcal{B}_2\mathbf{K}))^{-1}\boldsymbol{\delta}_x, \end{aligned} \tag{3-3}$$

or, when defining $\boldsymbol{\Phi}_x : \boldsymbol{\delta}_x \to \mathbf{x}$ and $\boldsymbol{\Phi}_u : \boldsymbol{\delta}_x \to \mathbf{u}$, in a short form as:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_x \\ \boldsymbol{\Phi}_u \end{bmatrix} \boldsymbol{\delta}_x. \tag{3-4}$$

These mappings are, due to the required causality, constructed with a similar block-lower-triangular form as $\mathbf{K}$, as shown in Eq. (3-5). They can be seen as a matrix representation of the convolution operator, so the elements in a particular row are numbered in reverse. When there is no noise or disturbance (i.e. $\boldsymbol{\delta}_x$ only contains x_0 and zeros), only the first block column of $\boldsymbol{\Phi}_x$ and $\boldsymbol{\Phi}_u$ is of interest. This is then denoted by $\boldsymbol{\Phi}_x^0$ and $\boldsymbol{\Phi}_u^0$, respectively.

$$\boldsymbol{\Phi}_x = \begin{bmatrix} \Phi_x^{0,0} & & & \\ \Phi_x^{1,1} & \Phi_x^{1,0} & & \\ \vdots & \ddots & \ddots & \\ \Phi_x^{T,T} & \dots & \Phi_x^{T,1} & \Phi_x^{T,0} \end{bmatrix}, \quad \boldsymbol{\Phi}_u = \begin{bmatrix} \Phi_u^{0,0} & & & \\ \Phi_u^{1,1} & \Phi_u^{1,0} & & \\ \vdots & \ddots & \ddots & \\ \Phi_u^{T,T} & \dots & \Phi_u^{T,1} & \Phi_u^{T,0} \end{bmatrix} \tag{3-5}$$

3-2-2 The System-Level Parameterisation And System-Level Constraints

The SLS framework optimises these maps to obtain the optimal control input. To guarantee that the maps from Eq. (3-4) follow Eq. (3-3), the System-Level Parameterisation (SLP) is added to the optimisation problem:

$$\begin{bmatrix} I - Z\mathcal{A} & -Z\mathcal{B}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_x \\ \boldsymbol{\Phi}_u \end{bmatrix} = I. \tag{3-6}$$

This constraint can easily be verified to follow Eq. (3-3):

$$\begin{aligned} \begin{bmatrix} I - Z\mathcal{A} & -Z\mathcal{B}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_x \\ \mathbf{K}\boldsymbol{\Phi}_x \end{bmatrix} &= I \\ (I - Z\mathcal{A} - Z\mathcal{B}_2\mathbf{K})\boldsymbol{\Phi}_x &= I \\ (I - Z(\mathcal{A} + \mathcal{B}_2\mathbf{K}))(I - Z(\mathcal{A} + \mathcal{B}_2\mathbf{K}))^{-1} &= I. \end{aligned}$$

Recall that SLS was originally developed to add sparsity constraints to these problems. These (or similar) constraints can be added to the SLS problem through System-Level Constraints (SLCs), where the closed-loop responses $\boldsymbol{\Phi}_x$ and $\boldsymbol{\Phi}_u$ have to lie in convex sets $\mathcal{S}^x$ and $\mathcal{S}^u$, respectively. State or input constraints can also be added by requiring the state or input to lie in a convex set $\mathcal{X}$ and $\mathcal{U}$ for every $\boldsymbol{\delta}_x$ in $\mathcal{D}$, a convex set containing all possible disturbances. Mathematically, these are most commonly represented as:

$$\begin{gathered} \boldsymbol{\Phi}_x \in \mathcal{S}^x, \quad \boldsymbol{\Phi}_u \in \mathcal{S}^u, \\ \boldsymbol{\Phi}_x\boldsymbol{\delta}_x \in \mathcal{X}, \; \boldsymbol{\Phi}_u\boldsymbol{\delta}_x \in \mathcal{U}, \; \forall \boldsymbol{\delta}_x \in \mathcal{D}. \end{gathered} \tag{3-7}$$

3-2-3 The System-Level Synthesis Problem

One can formulate the general optimal control problem in the SLS framework with objective function $g(\boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u)$ as follows:

$$\begin{aligned} \underset{\boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u}{\text{minimize}} \quad & g(\boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u) \\ \text{subject to} \quad & \text{SLP from Eq. (3-6)}, \\ & \text{SLCs from Eq. (3-7)}, \end{aligned}$$

where the controller $\mathbf{K}$ can afterwards be found using[1]:

$$\mathbf{K} = \mathbf{\Phi}_u \mathbf{\Phi}_x^{-1}.$$

The objective function can be used to create an $\mathcal{H}_\infty$ or $\mathcal{L}_1$ infinity control problem. For example, the $\mathcal{H}_\infty$ optimal control problem uses the following objective function:

$$g(\mathbf{\Phi}_x, \mathbf{\Phi}_u) = \left\| \begin{bmatrix} \mathcal{Q}^{\frac{1}{2}} & \\ & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} \right\|_{2 \to 2}^2,$$

where $\mathcal{Q}$ and $\mathcal{R}$ are defined as:

$$\mathcal{Q} := \begin{bmatrix} Q & & & \\ & \ddots & & \\ & & Q & \\ & & & Q_T \end{bmatrix}, \quad \mathcal{R} := \begin{bmatrix} R & & & \\ & \ddots & & \\ & & R & \\ & & & R_T \end{bmatrix}.$$

However, a common assumption is that no noise is acting on the system. This means $\boldsymbol{\delta}_x$ is known beforehand and has the form:

$$\boldsymbol{\delta}_x = \begin{bmatrix} x_0^\mathsf{T} & 0 & \dots & 0 \end{bmatrix}^\mathsf{T}.$$

which allows for a simplification of this problem. Using Eq. (3-4) it becomes clear that only the first block column of $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_u$ are relevant without any noise acting on the system. Then, by defining $\mathbf{\Phi}_x^0$, $\mathbf{\Phi}_u^0$ and I^0 as follows:

$$\mathbf{\Phi}_x^0 := \begin{bmatrix} \Phi_x^{0,0} \\ \vdots \\ \Phi_x^{T,T} \end{bmatrix}, \quad \mathbf{\Phi}_u^0 := \begin{bmatrix} \Phi_u^{0,0} \\ \vdots \\ \Phi_u^{T,T} \end{bmatrix}, \quad I^0 := \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

the SLS problem can be simplified to:

$$\begin{aligned} \underset{\mathbf{\Phi}_x^0, \mathbf{\Phi}_u^0}{\text{minimize}} \quad & x_0^\mathsf{T} {\mathbf{\Phi}_x^0}^\mathsf{T} \mathcal{Q} \mathbf{\Phi}_x^0 x_0 + x_0^\mathsf{T} {\mathbf{\Phi}_u^0}^\mathsf{T} \mathcal{R} \mathbf{\Phi}_u^0 x_0 \\ \text{subject to} \quad & \begin{bmatrix} I - Z\mathcal{A} & -Z\mathcal{B}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x^0 \\ \mathbf{\Phi}_u^0 \end{bmatrix} = I^0, \\ & \mathbf{\Phi}_x^0 \in \mathcal{S}^x, \quad \mathbf{\Phi}_u^0 \in \mathcal{S}^u, \\ & \mathbf{\Phi}_x^0 x_0 \in \mathcal{X}, \quad \mathbf{\Phi}_u^0 x_0 \in \mathcal{U}. \end{aligned}$$

3-3 Robust System-Level Synthesis

To discuss the lumped SLS formulation, the classical robust SLS problem is presented first in Section 3-3-1. This is followed by the original lumped SLS formulation in Section 3-3-2, and an improved variant in Section 3-3-3.

[1]The inverse of $\mathbf{\Phi}_x$ always exists, as it is a lower triangular block matrix with identity matrices on the diagonal (see Eq. (3-3)). However, one may want to avoid computing the inverse of $\mathbf{\Phi}_x$ for large systems or large time horizons due to its size. One can then implement the controller in a feedback interconnection [29].

3-3-1 Classical Robust System-Level Synthesis

In practice, the exact system dynamics are often unknown, and the system can be affected by disturbances, which can lead to violations of the constraints. Assume that instead of the true dynamics $(A_t, B_{2.t})$, only the estimated dynamics $(\hat{A}_t, \hat{B}_{2,t})$ are available. This leads to the following dynamics:

$$x_{t+1} = \hat{A}_t x_t + \hat{B}_{2,t} u_t + (A_t - \hat{A}_t) x_t + (B_{2,,t} - \hat{B}_{2,t}) u_t + \delta_{x,t}.$$

With the estimated dynamics $(\hat{A}, \hat{B}_2)$ one can construct the matrices $(\hat{\mathcal{A}}, \hat{\mathcal{B}}_2)$ similarly to Eq. (3-2). The corresponding system responses $(\hat{\mathbf{\Phi}}_x, \hat{\mathbf{\Phi}}_u)$ then satisfy the state-feedback SLP:

$$\begin{bmatrix} I - Z\hat{\mathcal{A}} & -Z\hat{\mathcal{B}}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Phi}}_x \\ \hat{\mathbf{\Phi}}_u \end{bmatrix} = I.$$

The resulting controller found from $\hat{\mathbf{K}} = \hat{\mathbf{\Phi}}_u \hat{\mathbf{\Phi}}_x^{-1}$, however, is applied to the true dynamics $(\mathcal{A}, \mathcal{B}_2)$. This results in the following SLP with uncertain dynamics:

$$\begin{bmatrix} I - Z\mathcal{A} & -Z\mathcal{B}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Phi}}_x \\ \hat{\mathbf{\Phi}}_u \end{bmatrix} = I + Z \begin{bmatrix} \hat{\mathcal{A}} - \mathcal{A} & \hat{\mathcal{B}}_2 - \mathcal{B}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Phi}}_x \\ \hat{\mathbf{\Phi}}_u \end{bmatrix},$$

where the last term is defined as $\mathbf{\Delta}$:

$$\mathbf{\Delta} := Z \begin{bmatrix} \hat{\mathcal{A}} - \mathcal{A} & \hat{\mathcal{B}}_2 - \mathcal{B}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Phi}}_x \\ \hat{\mathbf{\Phi}}_u \end{bmatrix}.$$

This term has the following effect on $\mathbf{x}$ and $\mathbf{u}$:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{\Phi}}_x \\ \hat{\mathbf{\Phi}}_u \end{bmatrix} (I + \mathbf{\Delta})^{-1} \boldsymbol{\delta}_x,$$

which is used in the more classical robust SLS problems, where the goal is to synthesise a controller that performs well for different (and unknown) true systems dynamics $(\mathcal{A}, \mathcal{B}_2)$ with model uncertainty bounds $\epsilon_{\mathcal{A}}$ and $\epsilon_{\mathcal{B}_2}$:

$$\begin{aligned} \min_{\hat{\mathbf{\Phi}}_x, \hat{\mathbf{\Phi}}_u} \max_{\mathcal{A}, \mathcal{B}_2} \quad & \left\| \begin{bmatrix} \mathcal{Q}^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Phi}}_x \\ \hat{\mathbf{\Phi}}_u \end{bmatrix} (I + \mathbf{\Delta})^{-1} \right\|_F^2 \\ \text{subject to} \quad & \begin{bmatrix} I - Z\hat{\mathcal{A}} & -Z\hat{\mathcal{B}}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Phi}}_x \\ \hat{\mathbf{\Phi}}_u \end{bmatrix} = I, \\ & \left\| \hat{\mathcal{A}} - \mathcal{A} \right\|_{2\to2} \leq \epsilon_{\mathcal{A}}, \\ & \left\| \hat{\mathcal{B}}_2 - \mathcal{B}_2 \right\|_{2\to2} \leq \epsilon_{\mathcal{B}_2}, \end{aligned}$$

where F denotes the Frobenius norm.

3-3-2 Classical Lumped System-Level Synthesis

The classical robust SLS approach becomes quasi-convex at best [34], which led to the development of the lumped SLS variant in [28]. This method, the Classical Lumped System-Level Synthesis (CLSLS), is presented in four parts: the lumped uncertainties, bounding these uncertainties, tightening the constraints accordingly and then combining everything for the CLSLS problem.

Lumped Uncertainties

Uncertainties due to disturbances and due to model errors are grouped as one lumped uncertainty η_t:

$$\begin{aligned} x_{t+1} &= \hat{A}_t x_t + \hat{B}_{2,t} u_t + (A_t - \hat{A}_t)x_t + (B_{2,t} - \hat{B}_{2,t})u_t + \delta_{x,t} \\ &= \hat{A}_t x_t + \hat{B}_{2,t} u_t + \eta_t. \end{aligned} \tag{3-8}$$

The lumped uncertainty can also be expressed over time using signals, where it is important to note that, as η_t is now the disturbance, $\hat{\mathbf{\Phi}}_x : \boldsymbol{\eta} \to \boldsymbol{x}$ and $\hat{\mathbf{\Phi}}_u : \boldsymbol{\eta} \to \boldsymbol{u}$:

$$\begin{aligned} \boldsymbol{\eta} &= Z(\mathcal{A} - \hat{\mathcal{A}})\mathbf{x} + Z(\mathcal{B}_2 - \hat{\mathcal{B}}_2)\mathbf{u} + \boldsymbol{\delta}_x, \\ &= Z\begin{bmatrix} \mathcal{A} - \hat{\mathcal{A}} & \mathcal{B}_2 - \hat{\mathcal{B}}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Phi}}_x \\ \hat{\mathbf{\Phi}}_u \end{bmatrix} \boldsymbol{\eta} + \boldsymbol{\delta}_x, \end{aligned} \tag{3-9}$$

where the first entry of $\boldsymbol{\eta}$ is once again x_0.

Lumped Uncertainty Bound

The robust problem requires bounds on η_t, ideally as tight as possible to reduce conservativeness. However, this is not an easy task due to its dependence on the states and inputs. To bound the uncertainty, the lumped uncertainty can equivalently be denoted as:

$$\eta_t = \sigma_t \, \tilde{\delta}_{x,t}. \tag{3-10}$$

where σ_t is a positive scalar and $\tilde{\delta}_{x,t}$ is noise such that $\left\|\tilde{\delta}_{x,t}\right\|_\infty \leq 1$. This means σ_t provides an upper bound on the infinity norm of the lumped disturbance:

$$\begin{aligned} \|\eta_t\|_\infty &\leq \sigma_t \, \left\|\tilde{\delta}_{x,t}\right\|_\infty, \\ &\leq \sigma_t. \end{aligned}$$

To bind the lumped uncertainty, it first has to be related to state and input variables. However, the formulation in Eq. (3-9) depends on the lumped uncertainty as well, whose bound is the objective to obtain in the first place. This chicken-egg problem can be removed by using the signal equivalent of Eq. (3-10) in Eq. (3-9):

$$\boldsymbol{\eta} = Z\begin{bmatrix} \mathcal{A} - \hat{\mathcal{A}} & \mathcal{B}_2 - \hat{\mathcal{B}}_2 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{\Phi}}_x \\ \tilde{\mathbf{\Phi}}_u \end{bmatrix} \tilde{\boldsymbol{\delta}}_x + \boldsymbol{\delta}_x, \tag{3-11}$$

where $\tilde{\mathbf{\Phi}}_x : \tilde{\boldsymbol{\delta}}_x \to \mathbf{x}$ and $\tilde{\mathbf{\Phi}}_u : \tilde{\boldsymbol{\delta}}_x \to \mathbf{u}$ are defined as:

$$\begin{aligned} \tilde{\mathbf{\Phi}}_x &:= \hat{\mathbf{\Phi}}_x \mathbf{\Sigma}, \\ \tilde{\mathbf{\Phi}}_u &:= \hat{\mathbf{\Phi}}_u \mathbf{\Sigma}, \end{aligned} \tag{3-12}$$

and where $\mathbf{\Sigma}$ and $\tilde{\boldsymbol{\delta}}_x$ follow from:

$$\mathbf{\Sigma} := \begin{bmatrix} I & & & \\ & \sigma_0 I & & \\ & & \ddots & \\ & & & \sigma_{T-1} I \end{bmatrix}, \quad \tilde{\boldsymbol{\delta}}_x := \begin{bmatrix} x_0 \\ \tilde{\delta}_{x,0} \\ \vdots \\ \tilde{\delta}_{x,T-1} \end{bmatrix}.$$

When Eq. (3-11) is formulated in its scalar form for an LTI system:

$$\begin{aligned} \eta_0 &= (A - \hat{A})\tilde{\Phi}_x^{0,0} x_0 + (B_2 - \hat{B}_2)\tilde{\Phi}_u^{0,0} x_0 + \delta_{x,0}, \\ \eta_t &= (A - \hat{A})(\tilde{\Phi}_x^{t,t} x_0 + \sum_{i=1}^{t} \tilde{\Phi}_x^{t,t-i} \tilde{\delta}_{x,i-1}) \\ &\qquad + (B_2 - \hat{B}_2)(\tilde{\Phi}_u^{t,t} x_0 + \sum_{i=1}^{t} \tilde{\Phi}_u^{t,t-i} \tilde{\delta}_{x,i-1}) + \delta_{x,t}, \quad \forall t \in \mathbb{Z}_1^{T-1}, \end{aligned} \tag{3-13}$$

it is possible to find the following upper bound on $\|\eta_t\|_\infty$ using the triangle inequality and the submultiplicative property of the infinity norm:

$$\begin{aligned} \|\eta_0\|_\infty &\leq \epsilon_A \left\| \tilde{\Phi}_x^{0,0} x_0 \right\|_\infty + \epsilon_{B_2} \left\| \tilde{\Phi}_u^{0,0} x_0 \right\|_\infty + \sigma_\delta \leq \sigma_0, \\ \|\eta_t\|_\infty &\leq \epsilon_A \left(\left\| \tilde{\Phi}_x^{t,t} x_0 \right\|_\infty + \sum_{i=1}^{t} \left\| \tilde{\Phi}_x^{t,t-i} \right\|_{\infty\to\infty} \right) \\ &\qquad + \epsilon_{B_2} \left(\left\| \tilde{\Phi}_u^{t,t} x_0 \right\|_\infty + \sum_{i=1}^{t} \left\| \tilde{\Phi}_u^{t,t-i} \right\|_{\infty\to\infty} \right) + \sigma_\delta \leq \sigma_t, \quad \forall t \in \mathbb{Z}_1^{T-1}. \end{aligned} \tag{3-14}$$

where $\left\|A - \hat{A}\right\|_{\infty\to\infty} \leq \epsilon_A$, $\left\|B_2 - \hat{B}_2\right\|_{\infty\to\infty} \leq \epsilon_{B_2}$ and $\|\delta_{x,t}\|_\infty \leq \sigma_\delta$, $\forall t$.

With the new closed-loop transfer matrices $\tilde{\mathbf{\Phi}}_x$ and $\tilde{\mathbf{\Phi}}_u$, the SLP changes as these transfer matrices do not map the disturbance $\boldsymbol{\eta}$ to the states or inputs, respectively. Where the SLP for the lumped disturbance $\boldsymbol{\eta}$ with $\hat{\mathbf{\Phi}}_x$ and $\hat{\mathbf{\Phi}}_u$ would have the following form:

$$\begin{bmatrix} I - Z\hat{\mathcal{A}} & -Z\hat{\mathcal{B}}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Phi}}_x \\ \hat{\mathbf{\Phi}}_u \end{bmatrix} = I, \tag{3-15}$$

the SLP with $\tilde{\mathbf{\Phi}}_x$ and $\tilde{\mathbf{\Phi}}_u$ has $\mathbf{\Sigma}$ on the right-hand side:

$$\begin{bmatrix} I - Z\hat{\mathcal{A}} & -Z\hat{\mathcal{B}}_2 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{\Phi}}_x \\ \tilde{\mathbf{\Phi}}_u \end{bmatrix} = \mathbf{\Sigma}. \tag{3-16}$$

Using Eq. (3-12) and Eq. (3-15), obtaining this result is trivial. Right multiplying both sides with $\tilde{\boldsymbol{\delta}}_x$ also yields the signal equivalent of Eq. (3-8).

Constraint Tightening

With these bounds for the magnitude of the lumped disturbance found, the next step is to tighten the constraint to prevent any constraint violations. The constraints are assumed to be of the form:

$$\begin{aligned} H_{\mathcal{X}_t} x_t &\leq h_{\mathcal{X}_t}, \\ H_{\mathcal{X}_T} x_T &\leq h_{\mathcal{X}_T}, \\ H_{\mathcal{U}_t} u_t &\leq h_{\mathcal{U}_t}. \end{aligned}$$

Furthermore, using that:

$$x_t = \tilde{\Phi}_x^{t,t} x_0 + \sum_{i=1}^{t} \tilde{\Phi}_x^{t,t} \tilde{\delta}_{i-1},$$

and using Hölder's inequality, the state constraints can be expressed as:

$$\begin{aligned} H^j_{\mathcal{X}_t} x_t &= H^j_{\mathcal{X}_t}\left(\tilde{\Phi}^{t,t}_x x_0 + \sum_{i=1}^{t} \tilde{\Phi}^{t,t}_x \tilde{\delta}_{i-1}\right) \\ &\leq H^j_{\mathcal{X}_t}\tilde{\Phi}^{t,t}_x x_0 + \sum_{i=1}^{t}\left\|H^j_{\mathcal{X}_t}\tilde{\Phi}^{t,t}_x\right\|_1 \left\|\tilde{\delta}_{i-1}\right\|_\infty \\ &\leq H^j_{\mathcal{X}_t}\tilde{\Phi}^{t,t}_x x_0 + \sum_{i=1}^{t}\left\|H^j_{\mathcal{X}_t}\tilde{\Phi}^{t,t}_x\right\|_1 \leq h^j_{\mathcal{X}_t}, \end{aligned}$$

with j denoting the j'th constraint (i.e., row) of $H_{\mathcal{X}_t}$ and $h_{\mathcal{X}_t}$. Furthermore, when looking at Eq. (3-16), it can be seen that:

$$\tilde{\Phi}^{t,0}_x = \sigma_{t-1} I, \quad \forall t \in \mathbb{Z}^T_1,$$

such that the tightened constraints can be rewritten as follows for each of the $n_{\mathcal{X}_t}$ constraints:

$$H^j_{\mathcal{X}_t}\tilde{\Phi}^{t,t}_x x_0 + \sum_{i=1}^{t-1}\left\|H^j_{\mathcal{X}_t}\tilde{\Phi}^{t,t-i}_x\right\|_1 + \sigma_{t-1}\left\|H^j_{\mathcal{X}_t}\right\|_1 \leq h^j_{\mathcal{X}_t},\ \forall j \in \mathbb{Z}^{n_{\mathcal{X}_t}-1}_0,\ \forall t \in \mathbb{Z}^{T-1}_1. \tag{3-17}$$

Following a similar procedure, similar equations are obtained for $\mathcal{X}_T$ and $\mathcal{U}_t$:

$$H^j_{\mathcal{X}_T}\tilde{\Phi}^{T,T}_x x_0 + \sum_{i=1}^{T-1}\left\|H^j_{\mathcal{X}_T}\tilde{\Phi}^{T,T-i}_x\right\|_1 + \sigma_{T-1}\left\|H^j_{\mathcal{X}_T}\right\|_1 \leq h^j_{\mathcal{X}_T},\ \forall j \in \mathbb{Z}^{n_{\mathcal{X}_T}-1}_0, \tag{3-18}$$

$$H^j_{\mathcal{U}_t}\tilde{\Phi}^{t,t}_u x_0 + \sum_{i=1}^{t}\left\|H^j_{\mathcal{U}_t}\tilde{\Phi}^{t,t-i}_u\right\|_1 \leq h^j_{\mathcal{U}_t},\ \forall j \in \mathbb{Z}^{n_{\mathcal{U}_t}-1}_0,\ \forall t \in \mathbb{Z}^{T-1}_0. \tag{3-19}$$

As expected, the estimated lumped uncertainty decreases the feasible region for the state constraints.

Classical Lumped System-Level Synthesis Problem

With all these expressions combined, it is possible to formulate the CLSLS problem:

$$\begin{aligned} \underset{\tilde{\mathbf{\Phi}}_x,\, \tilde{\mathbf{\Phi}}_u,\, \mathbf{\Sigma}}{\text{minimize}} \quad & x_0^\top {\mathbf{\Phi}^0_x}^\top \mathcal{Q} \mathbf{\Phi}^0_x x_0 + x_0^\top {\mathbf{\Phi}^0_u}^\top \mathcal{R} \mathbf{\Phi}^0_u x_0 \\ \text{subject to} \quad & \begin{bmatrix} I - Z\hat{\mathcal{A}} & -Z\hat{\mathcal{B}}_2 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{\Phi}}_x \\ \tilde{\mathbf{\Phi}}_u \end{bmatrix} = \mathbf{\Sigma}, \\ & \text{Lumped uncertainty bounds from Eq. (3-14)}, \\ & \text{Tightened constraints from Eqs. (3-17) to (3-19)}. \end{aligned}$$

3-3-3 Improved Lumped System-Level Synthesis

As the CLSLS approach from [28] overapproximates the infinity-norm of the lumped disturbance, the worst-case scenario of the most uncertain state will also limit a possibly uncoupled and perfectly modelled state. This shared conservativeness can limit the performance of the controller, which is addressed with the Improved Lumped System-Level Synthesis (ILSLS). The basics are the same as the CLSLS, but the differences between the models are discussed in three parts: the lumped uncertainty bound, the tightened constraints and finally the problem formulation.

Lumped Uncertainty Bound

Instead of overapproximating the magnitude of every η_t with a single scalar σ_t, every state j in η_t is individually overapproximated with its own σ_t^j. Thus, instead of Eq. (3-10), η_t is approximated as:

$$\eta_t = \begin{bmatrix} \sigma_t^0 & & \\ & \ddots & \\ & & \sigma_t^{n_x-1} \end{bmatrix} \tilde{\delta}_{x,t}.$$

$\mathbf{\Sigma}$ is build out of the matrices Σ_t, which are both defined as:

$$\mathbf{\Sigma} := \begin{bmatrix} I & & & \\ & \Sigma_0 & & \\ & & \ddots & \\ & & & \Sigma_{T-1} \end{bmatrix}, \quad \Sigma_t := \begin{bmatrix} \sigma_t^0 & & \\ & \ddots & \\ & & \sigma_t^{n_x-1} \end{bmatrix}.$$

This formulation is not new, as it is also used in [35] and [36]. However, both of these approaches scale poorly when the number of sources of uncertainty grows. That is, if $A - \hat{A}$ is taken as an example, two possible model uncertainties are provided in a small example:

$$A - \hat{A} = \begin{bmatrix} \epsilon_A^0 & \epsilon_A^2 & 2\epsilon_A^0 \\ 2\epsilon_A^1 & 5\epsilon_A^2 & \epsilon_A^2 \\ \epsilon_A^1 & 7\epsilon_A^0 & 9\epsilon_A^1 \end{bmatrix}, \quad A - \hat{A} = \begin{bmatrix} \epsilon_A^0 & \epsilon_A^3 & \epsilon_A^6 \\ \epsilon_A^1 & \epsilon_A^4 & \epsilon_A^7 \\ \epsilon_A^2 & \epsilon_A^5 & \epsilon_A^8 \end{bmatrix}.$$

In the leftmost example, there are three different sources of uncertainty (i.e., ϵ_A^0, ϵ_A^1 and ϵ_A^2). There is some underlying structure in this case, which the work in [35] and [36] uses to robustly control a system.

In the rightmost example, however, there is no structure in the uncertainty and nine different sources of uncertainty are present. The number of constraints in [35] and [36] scale exponentially with the number of sources of uncertainty, which shows that without any assumptions a new formulation must be developed for larger systems.

Therefore, in this work, a formulation is provided for norm-bounded model uncertainties for general norm-bounded disturbances. To do so, first an upper bound on the magnitude of the lumped disturbance similar to Eq. (3-14) is found. Starting from Eq. (3-13), it is possible to find an upper bound of the magnitude of state j. Starting with η_0, this yields:

$$\begin{aligned} \left|\eta_0^j\right| &= \left|(A^j - \hat{A}^j)\tilde{\Phi}_x^{0,0}x_0 + (B_2^j - \hat{B}_2^j)\tilde{\Phi}_u^{0,0}x_0 + \delta_{x,0}\right| \\ &\leq \epsilon_A^j \left\|\tilde{\Phi}_x^{0,0}x_0\right\|_\infty + \epsilon_{B_2}^j \left\|\tilde{\Phi}_u^{0,0}x_0\right\|_\infty + \sigma_\delta^j \leq \sigma_0^j, \quad \forall j \in \mathbb{Z}_0^{n_x-1}, \end{aligned} \tag{3-20}$$

where $(\cdot)^j$ denotes the j'th row of the corresponding matrix or vector, and where ϵ_A^j, $\epsilon_{B_2}^j$ and σ_δ^j are defined as:

$$\epsilon_A^j := \left\|A^j - \hat{A}^j\right\|_1, \quad \epsilon_{B_2}^j := \left\|B_2^j - \hat{B}_2^j\right\|_1, \quad \sigma_\delta^j := \max |\delta_{x,t}^j|, \quad \forall t.$$

Following a similar procedure, the bound for η_t^j can be found:

$$\begin{aligned} |\eta_t^j| \leq \epsilon_A^j &\left(\left\|\tilde{\Phi}_x^{t,t}x_0\right\|_\infty + \sum_{i=1}^{t} \left\|\tilde{\Phi}_x^{t,t-i}\right\|_{\infty\to\infty} \right) \\ &+ \epsilon_{B_2}^j \left(\left\|\tilde{\Phi}_u^{t,t}x_0\right\|_\infty + \sum_{i=1}^{t} \left\|\tilde{\Phi}_u^{t,t-i}\right\|_{\infty\to\infty} \right) + \sigma_\delta^j \leq \sigma_t^j, \quad \forall j \in \mathbb{Z}_0^{n_x-1}, \quad \forall t \in \mathbb{Z}_1^{T-1}. \end{aligned} \tag{3-21}$$

Constraint Tightening

The different parameterisation of $\mathbf{\Sigma}$ has a small effect on the tightened constraints of $\mathcal{X}_t$ and $\mathcal{X}_T$, as $\tilde{\Phi}_x^{t,0}$ is not equal to $\sigma_{t-1}I$, but now equal to Σ_{t-1}. For completeness, the constraints for $\mathcal{U}_t$ are also provided, although these have remained the same:

$$
\begin{aligned}
H^j_{\mathcal{X}_t}\tilde{\Phi}_x^{t,t}x_0 + \sum_{i=1}^{t-1}\left\|H^j_{\mathcal{X}_t}\tilde{\Phi}_x^{t,t-i}\right\|_1 + \left\|H^j_{\mathcal{X}_t}\Sigma_{t-1}\right\|_1 &\leq h^j_{\mathcal{X}_t},\ \forall j \in \mathbb{Z}_0^{n_{\mathcal{X}_t}-1},\ \forall t \in \mathbb{Z}_1^{T-1}, \\
H^j_{\mathcal{X}_T}\tilde{\Phi}_x^{T,T}x_0 + \sum_{i=1}^{T-1}\left\|H^j_{\mathcal{X}_T}\tilde{\Phi}_x^{T,T-i}\right\|_1 + \left\|H^j_{\mathcal{X}_T}\Sigma_{T-1}\right\|_1 &\leq h^j_{\mathcal{X}_T},\ \forall j \in \mathbb{Z}_0^{n_{\mathcal{X}_T}-1}, \\
H^j_{\mathcal{U}_t}\tilde{\Phi}_u^{t,t}x_0 + \sum_{i=1}^{t}\left\|H^j_{\mathcal{U}_t}\tilde{\Phi}_u^{t,t-i}\right\|_1 &\leq h^j_{\mathcal{U}_t},\ \forall j \in \mathbb{Z}_0^{n_{\mathcal{U}_t}-1},\ \forall t \in \mathbb{Z}_0^{T-1}.
\end{aligned}
\tag{3-22}
$$

Improved Lumped System-Level Synthesis Problem

Combing these new equations leads to the following ILSLS problem:

$$
\begin{aligned}
\underset{\tilde{\mathbf{\Phi}}_x,\, \tilde{\mathbf{\Phi}}_u,\, \mathbf{\Sigma}}{\text{minimize}} \quad & x_0^\mathsf{T}{\mathbf{\Phi}_x^0}^\mathsf{T}\mathcal{Q}\mathbf{\Phi}_x^0 x_0 + x_0^\mathsf{T}{\mathbf{\Phi}_u^0}^\mathsf{T}\mathcal{R}\mathbf{\Phi}_u^0 x_0 \\
\text{subject to} \quad & \begin{bmatrix} I - Z\hat{\mathcal{A}} & -Z\hat{\mathcal{B}}_2 \end{bmatrix}\begin{bmatrix}\tilde{\mathbf{\Phi}}_x \\ \tilde{\mathbf{\Phi}}_u\end{bmatrix} = \mathbf{\Sigma}, \\
& \text{Lumped uncertainty bounds from Eqs. (3-20) and (3-21),} \\
& \text{Tightened constraints from Eq. (3-22).}
\end{aligned}
\tag{3-23}
$$

3-4 Controller Comparison

The results are split up into two sections. First, a simulation without disturbances shows the effect of accounting for the model inaccuracies is tested in Section 3-4-1. Secondly, a simulation with model uncertainties and disturbances is shown in Section 3-4-2.

3-4-1 Model Uncertainty Only

To show the effect of including these model uncertainties, a simulation is run without any disturbances first. The control parameters for this simulation are provided in Appendix B-2, where compared to the parameters for Section 2-3, the prediction horizon is decreased, and the terminal cost is increased. The model uncertainties are estimated using the procedure explained in Appendix D-1. General parameters for the scenario are provided in Table 3-1.

Table 3-1: Parameters during the first robustness scenario.

Symbol	Meaning	Value
a	Semi-major axis	55 m
e	Eccentricity	0
i	Inclination	45 deg
ω	Argument of periapsis	0 deg
Ω	RAAN	0 deg
N_{sat}	Number of satellites in the simulation	1 satellite

The results for the main states are shown in Fig. 3-1. Two things stand out with these results:

- The ILSLS formulation is significantly less conservative than the CLSLS formulation. Where the new formulation closely tracks the nominal SLS formulation, the original formulation limits the radial error to guarantee staying within bounds.
- The differences for the angular state $\delta\theta$ and the out-of-plane state $\delta\Omega$ are significantly smaller. Especially for the latter, almost no differences are visible, which is also visible when different models were compared in Section 2-3-5.

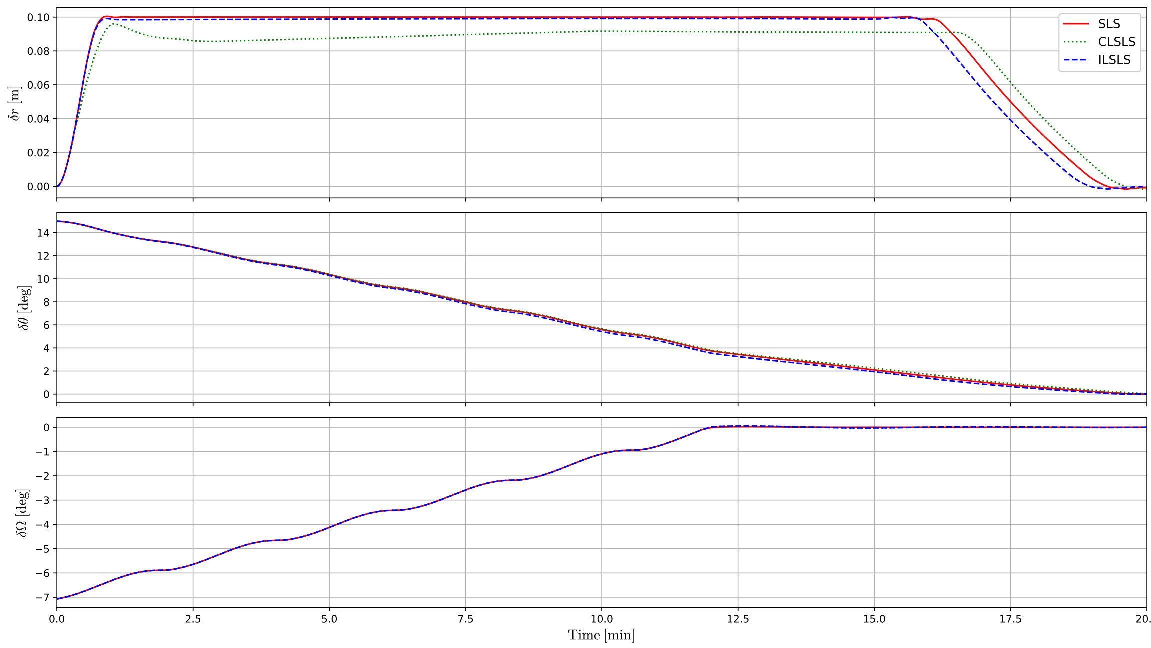


Figure 3-1: Main states for robust scenario without disturbances.

To better see the effect of the lumped SLS controller, a close-up view of the radial error is shown in Fig. 3-2. It is clear that the nominal SLS controller exceeds the limit of 0.1 m, not only during the initial overshoot but also afterwards. The controller is aware of this error during the simulation and tries to make a small correction to steer the satellite back to within the feasible region. However, because of the model inaccuracies, it fails to do so correctly.

The ILSLS formulation does properly stay within the required bounds. As the satellite approaches the reference and thus decreases the state variables, the ILSLS controller comes closer to the allowed limit. Similar behaviour is visible for the CLSLS controller in Fig. 3-1.

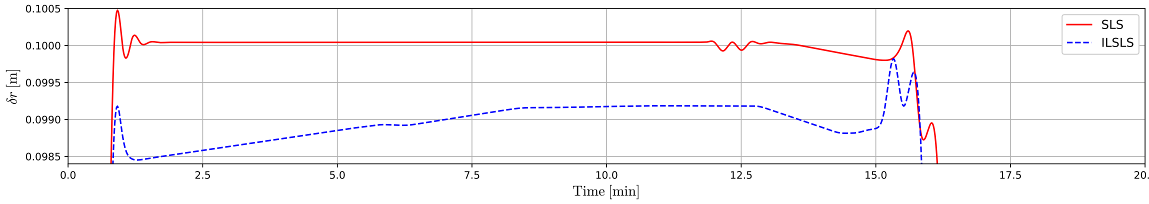


Figure 3-2: Close-up of radial state for robust scenario without disturbances.

The control inputs are visible in Fig. 3-3. The control inputs for most controllers are similar, although the nominal SLS controller uses less control input for u_r. The CLSLS controller differs from the other controllers at the start for u_t as well.

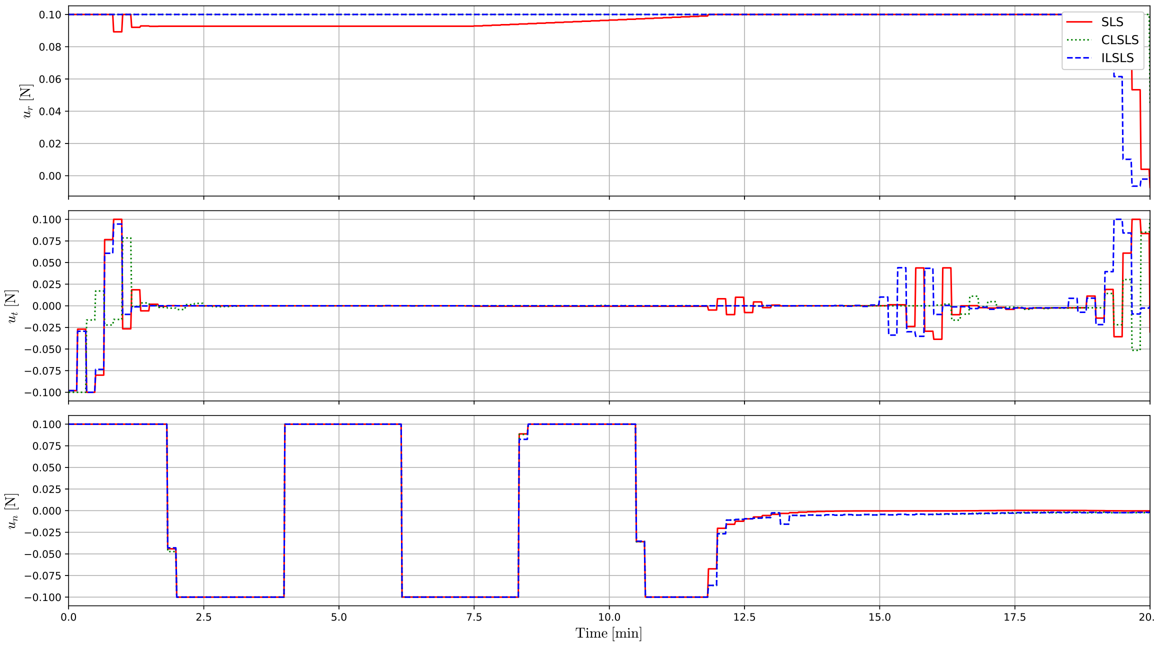


Figure 3-3: Control inputs for robust scenario without disturbances.

Upon closer inspection, it can seem weird that the new lumped SLS formulation in Fig. 3-1 is the first controller to decrease the state errors to zero. Usually, the conservativeness of a robust controller causes it to slow down compared to its non-robust counterpart, which is visible for the CLSLS controller, for example. However, in this particular scenario, by slightly decreasing the radial state δr, the controller can decrease its angular velocity further and therefore decrease $\delta\theta$ faster. Recall the equations of motion for the in-plane states from Section 2-2-3:

$$\begin{bmatrix} \delta\dot{r} \\ \delta\dot{\lambda}^f \\ \delta\dot{e}_x^f \\ \delta\dot{e}_y^f \end{bmatrix} = n_c \begin{bmatrix} r_c \delta e_y^f \\ -\frac{3}{2}\frac{1}{r_c}\delta r + \frac{1}{2}\delta e_x^f \\ -\delta e_y^f + \frac{2}{a_c n_c^2} u_t \\ \delta e_x^f + \frac{1}{a_c n_c^2} u_r \end{bmatrix}.$$

As the controller is nearing its radial limit, it has to guarantee a non-positive $\delta\dot{r}$, and thus a non-positive δe_y^f. This can be done through the control input u_r and the state variable δe_x^f.

The first large difference in the control inputs between SLS and ILSLS can be seen for the tangential input u_t at the fourth time step, where the lumped SLS takes a more conservative approach. This is required as due to the constraint tightening, the path and inputs planned by the nominal SLS controller are no longer possible, see Fig. 3-4.

In the following iterations, the nominal SLS controller provides larger control inputs for u_t (see Fig. 3-3), whereas the robust SLS controller, due to the forced change, stabilises at a lower value for δe_x^f as shown in Fig. 3-5. Both approaches decrease $\delta\dot{r}$, but the latter also decreases $\delta\dot{\lambda}^f$ further. This causes the lumped SLS controller to move faster than the nominal one.

However, if δr would decrease too much, this would increase $\dot{\lambda}_f$ again. This is why the CLSLS controller is slower, even though it can be seen in Fig. 3-5 that its value for δe_x^f is lower than the nominal controller several minutes.

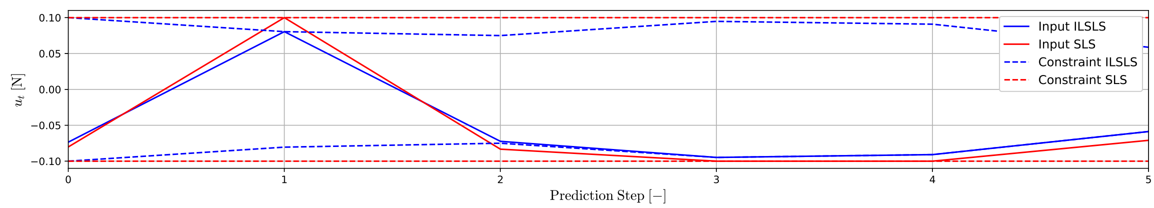


Figure 3-4: Comparison of inputs for both the nominal and robust controller at fourth iteration.

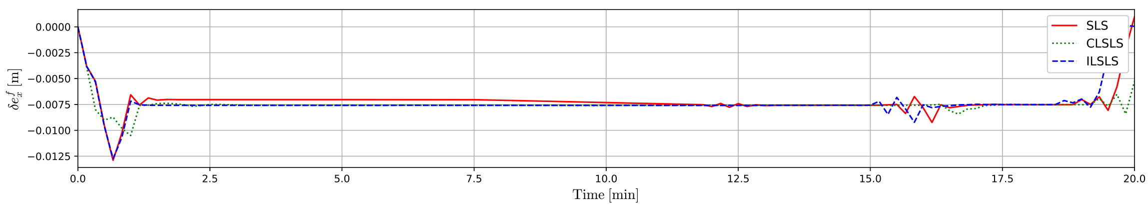


Figure 3-5: Comparison of state δe_x^f for robustness scenario.

Finally, it is important to address the metrics discussed in Section 2-3-1. These are shown in Table 3-2, where a large difference is seen for the computation times. The lumped controllers are more than 400 times slower than the nominal SLS controller. This can be explained by the different norms required for the lumped SLS formulation, which means standard quadratic solvers alone cannot be used anymore by Gurobi. For this small simulation with only a single satellite, the ILSLS controller is not slower than the CLSLS controller.

Table 3-2: Metrics during robust scenario without disturbances.

Model	$\|\bar{\mathbf{u}}\|_2$ [N]	$\|\bar{\mathbf{u}}\|_2^{\text{norm}}$ [-]	T_{sol} [s]	$T_{\text{sol}}^{\text{norm}}$ [-]
SLS	0.1227	1.0000	0.0015	1.0000
CLSLS	0.1253	1.0211	0.6299	424.4849
ILSLS	0.1231	1.0031	0.6265	422.2021

3-4-2 Model Uncertainty And Disturbance

Disturbances are added to the simulation with the procedure explained in Appendix D-2, where the effect on the state variables is estimated as well. With these disturbances added to the simulation, the differences between the different methods become increasingly apparent. When looking at Fig. 3-6 for example, the nominal SLS method moves towards the given radial limit and, due to model uncertainties and the disturbances, exceeds this limit several times.

The CLSLS problem accounts for the disturbances but has no way of distinguishing between a small disturbance on one state or a significant disturbance on another. Therefore, it assumed a large disturbance on all states, causing it to be incredibly conservative.

The ILSLS formulation is more conservative than the nominal SLS method, as it steers clear from the provided limit of 0.1 m. However, it is significantly less conservative than the original formulation and manages to once more arrive at the reference first.

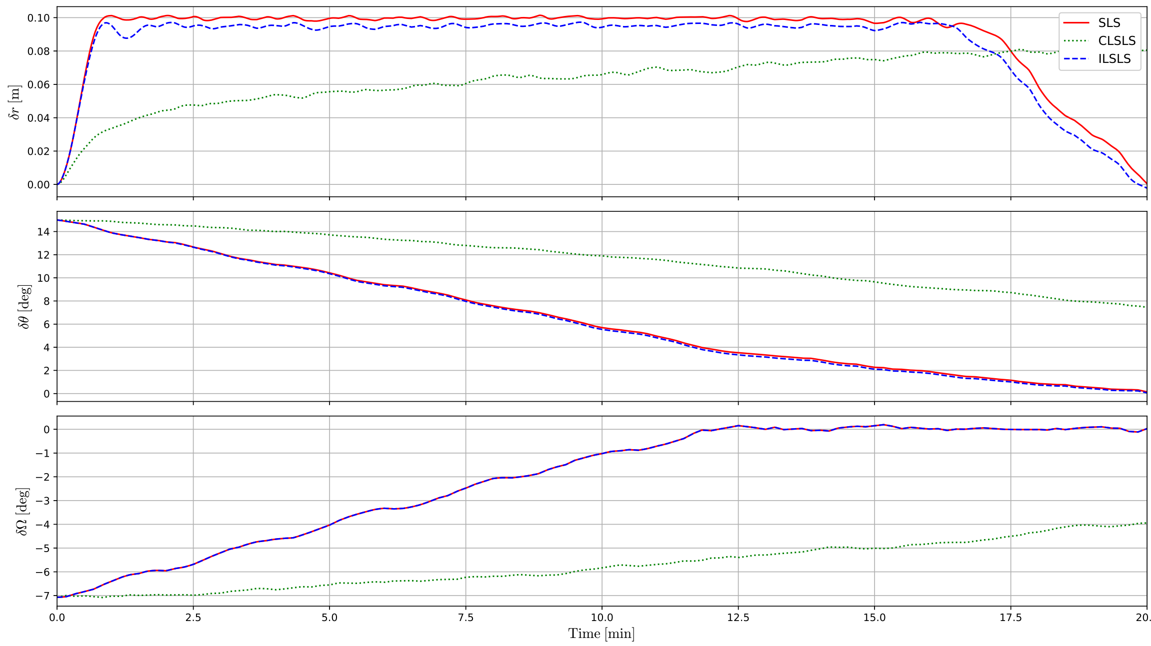


Figure 3-6: Main states for robust scenario with disturbances.

When looking at the control inputs in Fig. 3-7, it is clear that the CLSLS controller is the most conservative of the three. It reaches the maximum input value once for none of the three inputs. The newly proposed lumped SLS controller provides a constant maximum input for the radial input but shows very oscillatory behaviour for the tangential input.

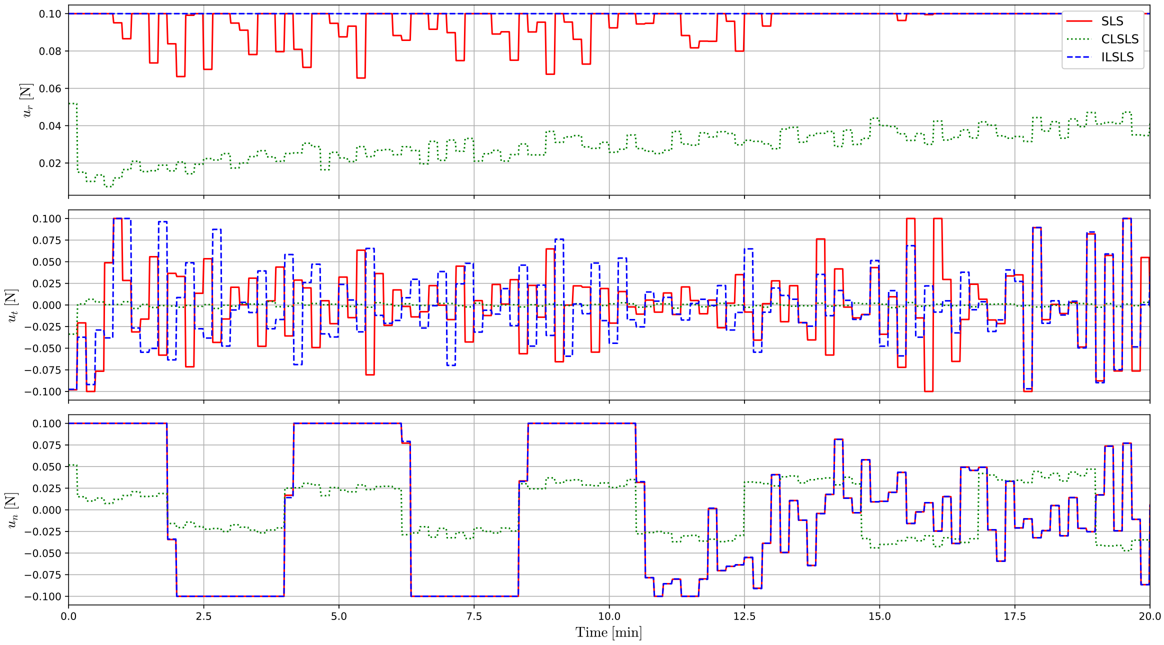


Figure 3-7: Control inputs for robust scenario with disturbances.

The metrics tell the same story as the figures above, and the computation time for the lumped SLS controllers is once more significantly higher than for the original SLS controller as shown in Table 3-3.

Table 3-3: Metrics during robust scenario with disturbances.

Model	$\|\bar{\mathbf{u}}\|_2$ [N]	$\|\bar{\mathbf{u}}\|_2^{\text{norm}}$ [-]	T_{sol} [s]	$T_{\text{sol}}^{\text{norm}}$ [-]
SLS	0.1300	3.0896	0.0015	1.0000
CLSLS	0.0421	1.0000	0.5796	390.6032
ILSLS	0.1329	3.1592	0.6143	413.9882

3-5 Conclusions

The goal of this chapter was to find a scalable, robust control method that can be used to move towards a given reference while being guaranteed to satisfy its constraints. Because of its promising computational speed and low conservativeness, the lumped SLS method of [28] was chosen.

This method is able to guarantee that the constraints are met. Still, due to the different scales of different variables and their uncertainties, this method is too conservative for the problem at hand. To solve this issue, a modification also used in [35] and [36] was applied, where the lumped uncertainty for each state variable individually is calculated. These existing methods require many constraints to model an unknown norm-bounded uncertainty, which is the reason a new formulation was developed.

This formulation has been shown to be considerably less conservative than the original lumped SLS formulation from [28]. Still, unlike the nominal SLS model, it is able to stay within the provided bounds at all times. It is significantly slower than nominal SLS, however, which is why, in the next chapter, the goal is to increase the computational speed of the algorithm.

Chapter 4

Optimisation

Implementing an efficient optimisation routine is a critical step in making the control approach scalable. This chapter discusses different formulations to optimise the time it takes several solvers to find the optimal solution. First the nominal SLS problem is analysed in Section 4-1, after which the robust SLS problem is improved in Section 4-2. These new problem formulations are tested for various solvers in Section 4-3, after which conclusions are drawn in Section 4-4.

4-1 Nominal System-Level Synthesis Variations

The goal is to implement a scalable robust SLS algorithm such that a large constellation can safely be controlled. However, as the optimisation problem as posed in Eq. (3-23) is complex, the optimisation problem is sped up through three separate steps:

1. The nominal SLS problem uses the matrices $\mathbf{\Phi}_x^0$ and $\mathbf{\Phi}_u^0$ as decision variables and is shortly presented in Section 4-1-1. However, not every solver can deal with matrix decision variables, and if they do, their selected algorithms might not be as efficient as those for vector decision variables. Therefore, the nominal SLS problem is transformed into a standard quadratic problem with vectors as decision variables in Section 4-1-2.

2. The decision variables $\mathbf{\Phi}_x^0$ and $\mathbf{\Phi}_u^0$ are often sparse, which means a significant number of entries of these matrices are zero. These entries can be left out of the optimisation problem, as they do not affect multiplications, divisions, additions and subtractions[1]. The sparse SLS formulation is derived in Section 4-1-3.

3. The robust SLS problem contains one-norms and infinity-norms that a standard quadratic problem cannot cope with. However, as shown in Section 4-2, it is possible to find an equivalent formulation that does fit the standard quadratic problem format.

4-1-1 Dense System-Level Synthesis

The standard and dense SLS problem optimises over the matrix variables $\mathbf{\Phi}_x^0$ and Φ_u^0. This problem formulation was derived in Section 3-2-2, although the constraints are slightly altered. The sparsity constraints for $\mathcal{S}^x$ and $\mathcal{S}^u$ are removed for the dense problem, and the state and

[1] All zero entries should, whether the matrix is sparse or dense, never occur as the sole denominator in a fraction, of course. Division, therefore, is no problem as long as the original problem was well-posed.

input constraints are defined as:

$$\begin{aligned} x_{\min} \leq & \mathbf{\Phi}_x^0 x_0 \leq x_{\max}, \\ u_{\min} \leq & \mathbf{\Phi}_u^0 x_0 \leq u_{\max}. \end{aligned}$$

This results in the following SLS problem:

$$\begin{aligned} \underset{\mathbf{\Phi}_x^0,\,\mathbf{\Phi}_u^0}{\text{minimize}} \quad & x_0^\mathsf{T} {\mathbf{\Phi}_x^0}^\mathsf{T} \mathcal{Q} \mathbf{\Phi}_x^0 x_0 + x_0^\mathsf{T} {\mathbf{\Phi}_u^0}^\mathsf{T} \mathcal{R} \mathbf{\Phi}_u^0 x_0 \\ \text{subject to} \quad & \begin{bmatrix} I - Z\mathcal{A} & -Z\mathcal{B}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x^0 \\ \mathbf{\Phi}_u^0 \end{bmatrix} = I^0, \\ & x_{\min} \leq \mathbf{\Phi}_x^0 x_0 \leq x_{\max}, \\ & u_{\min} \leq \mathbf{\Phi}_u^0 x_0 \leq u_{\max}. \end{aligned} \tag{4-1}$$

4-1-2 Transformed System-Level Synthesis

Not all solvers can deal with the required matrix decision variables from Eq. (4-1). To solve this problem, a transformed SLS problem is constructed where first, a vector version of the matrix variables must be defined. The different entries of, for example, $\Phi_x^{i,j}$ are labelled as follows, given that the total state size is n_x:

$$\Phi_x^{i,j} = \begin{bmatrix} \varphi_0^{x,i,j} & \varphi_{n_x}^{x,i,j} & \cdots & \varphi_{(n_x-2)n_x}^{x,i,j} & \varphi_{(n_x-1)n_x}^{x,i,j} \\ \varphi_1^{x,i,j} & \varphi_{n_x+1}^{x,i,j} & \cdots & \varphi_{(n_x-2)n_x+1}^{x,i,j} & \varphi_{(n_x-1)n_x+1}^{x,i,j} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \varphi_{n_x-2}^{x,i,j} & \varphi_{2n_x-2}^{x,i,j} & \cdots & \varphi_{(n_x-2)n_x-2}^{x,i,j} & \varphi_{n_x^2-2}^{x,i,j} \\ \varphi_{n_x-1}^{x,i,j} & \varphi_{2n_x-1}^{x,i,j} & \cdots & \varphi_{(n_x-2)n_x-1}^{x,i,j} & \varphi_{n_x^2-1}^{x,i,j} \end{bmatrix}, \tag{4-2}$$

of which the corresponding vector form is denoted by $\varphi_x^{i,j}$:

$$\varphi_x^{i,j} := \begin{bmatrix} \varphi_0^{x,i,j} & \varphi_1^{x,i,j} & \dots & \varphi_{n_x^2-2}^{x,i,j} & \varphi_{n_x^2-1}^{x,i,j} \end{bmatrix}^\mathsf{T}.$$

The vector $\boldsymbol{\varphi}_x^0$ (and similarly for $\boldsymbol{\varphi}_u^0$) contains all these $\varphi_x^{i,j}$ and can be seen as the vector equivalent of $\mathbf{\Phi}_x^0$:

$$\boldsymbol{\varphi}_x^0 = \begin{bmatrix} {\varphi_x^{0,0}}^\mathsf{T} & {\varphi_x^{1,1}}^\mathsf{T} & \dots & {\varphi_x^{T-1,T-1}}^\mathsf{T} & {\varphi_x^{T,T}}^\mathsf{T} \end{bmatrix}^\mathsf{T}.$$

By defining $\mathcal{K}_x$ and $\mathcal{K}_u$ as follows:

$$\mathcal{K}_x := I_{T+1} \otimes (x_0^\mathsf{T} \otimes I_{n_x}), \quad \mathcal{K}_u := I_{T+1} \otimes (x_0^\mathsf{T} \otimes I_{n_u}),$$

and $\mathcal{A}^\varphi$, $\mathcal{B}_2^\varphi$ and $\mathcal{I}^\varphi$ as:

$$\mathcal{A}^\varphi := \begin{bmatrix} I_{n_x} \otimes A & & \\ & \ddots & \\ & & I_{n_x} \otimes A \\ & & & 0 \end{bmatrix}, \mathcal{B}_2^\varphi := \begin{bmatrix} I_{n_x} \otimes B_2 & & \\ & \ddots & \\ & & I_{n_x} \otimes B_2 \\ & & & 0 \end{bmatrix}, \mathcal{I}^\varphi := \begin{bmatrix} I_{n_x}^{\text{flat}} \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where $I_{n_x}^{\text{flat}}$ denotes a flattened identity matrix, the transformed SLS problem formulation can be constructed:

$$\begin{aligned}
\underset{\boldsymbol{\varphi}_x^0,\boldsymbol{\varphi}_u^0}{\text{minimize}} \quad & \boldsymbol{\varphi}_x^{0\top} \mathcal{Q}^{\varphi} \boldsymbol{\varphi}_x^0 + \boldsymbol{\varphi}_u^{0\top} \mathcal{R}^{\varphi} \boldsymbol{\varphi}_u^0 \\
\text{subject to} \quad & \begin{bmatrix} I - Z\mathcal{A}^{\varphi} & -Z\mathcal{B}_2^{\varphi} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi}_x^0 \\ \boldsymbol{\varphi}_u^0 \end{bmatrix} = \mathcal{I}^{\varphi}, \\
& x_{\min} \leq \mathcal{K}_x \boldsymbol{\varphi}_x^0 \leq x_{\max}, \\
& u_{\min} \leq \mathcal{K}_u \boldsymbol{\varphi}_u^0 \leq u_{\max},
\end{aligned} \tag{4-3}$$

where $\mathcal{Q}^{\varphi}$ and $\mathcal{R}^{\varphi}$ are defined as:

$$\mathcal{Q}^{\varphi} := \mathcal{K}_x^\top \mathcal{Q} \mathcal{K}_x, \quad \mathcal{R}^{\varphi} := \mathcal{K}_u^\top \mathcal{R} \mathcal{K}_u.$$

The derivation of this equivalent formulation is provided in Appendix E-1. Afterwards, obtaining $\mathbf{\Phi}_x^0$ and $\mathbf{\Phi}_u^0$ following Eq. (4-2) is straightforward.

4-1-3 Sparse System-Level Synthesis

The construction of the transformed SLS problem in Section 4-1-2 has an extra benefit, namely that it is now straightforward to optimise sparse matrix variables $\mathbf{\Phi}_x^0$ and $\mathbf{\Phi}_u^0$. All entries of these matrices that either mathematically must be zero or, because the closed-loop transfer matrix is designed as such, can be left out in the transformed problem. Sparse closed-loop transfer matrices are obtained after the results are reconstructed from the closed-loop transfer matrices.

First the three different levels of sparsity are discussed, after which the corresponding indices are denoted. The new sparse problem formulation is presented last.

Levels of Sparsity

The sparsity in the closed-loop transfer matrices is present at three different levels: a micro, meso and macro level. The micro-level sparsity determines the coupling between different states. For the Blend model, a division is made between in-plane and out-of-plane states. For each satellite s, this structure for $\Phi_x^{i,j}$ and $\Phi_u^{i,j}$ is as follows:

$$\Phi_{x,s}^{i,j} = \begin{bmatrix} \varphi_0^{x,i,j} & \varphi_4^{x,i,j} & \varphi_8^{x,i,j} & \varphi_{12}^{x,i,j} & & \\ \varphi_1^{x,i,j} & \varphi_5^{x,i,j} & \varphi_9^{x,i,j} & \varphi_{13}^{x,i,j} & & \\ \varphi_2^{x,i,j} & \varphi_6^{x,i,j} & \varphi_{10}^{x,i,j} & \varphi_{14}^{x,i,j} & & \\ \varphi_3^{x,i,j} & \varphi_7^{x,i,j} & \varphi_{11}^{x,i,j} & \varphi_{15}^{x,i,j} & & \\ & & & & \varphi_{16}^{x,i,j} & \varphi_{18}^{x,i,j} \\ & & & & \varphi_{17}^{x,i,j} & \varphi_{19}^{x,i,j} \end{bmatrix}$$

$$\Phi_{u,s}^{i,j} = \begin{bmatrix} \varphi_0^{u,i,j} & \varphi_2^{u,i,j} & \varphi_4^{u,i,j} & \varphi_6^{u,i,j} & & \\ \varphi_1^{u,i,j} & \varphi_3^{u,i,j} & \varphi_5^{u,i,j} & \varphi_7^{u,i,j} & & \\ & & & & \varphi_8^{u,i,j} & \varphi_9^{u,i,j} \end{bmatrix}.$$

The meso-level sparsity provides the structure with respect to other satellites. Here, the simplest form is chosen, where each satellite finds its new state and inputs exclusively from its own previous state.

$$\Phi_x^{i,j} = \begin{bmatrix} \Phi_{x,0}^{i,j} & & \\ & \ddots & \\ & & \Phi_{x,N_{\text{sat}}}^{i,j} - 1 \end{bmatrix}, \quad \Phi_u^{i,j} = \begin{bmatrix} \Phi_{u,0}^{i,j} & & \\ & \ddots & \\ & & \Phi_{u,N_{\text{sat}}}^{i,j} - 1 \end{bmatrix}.$$

The macro-level sparsity provides the structure with respect to time. As the controller must be causal, this requires a block-lower-triangular matrix as shown in Eq. (3-5). This form is repeated here for convenience:

$$\mathbf{\Phi}_x = \begin{bmatrix} \Phi_x^{0,0} & & & \\ \Phi_x^{1,1} & \Phi_x^{1,0} & & \\ \vdots & \ddots & \ddots & \\ \Phi_x^{T,T} & \dots & \Phi_x^{T,1} & \Phi_x^{T,0} \end{bmatrix}, \quad \mathbf{\Phi}_u = \begin{bmatrix} \Phi_u^{0,0} & & & \\ \Phi_u^{1,1} & \Phi_u^{1,0} & & \\ \vdots & \ddots & \ddots & \\ \Phi_u^{T,T} & \dots & \Phi_u^{T,1} & \Phi_u^{T,0} \end{bmatrix}.$$

Sparsity Indices

The resulting sparsity requires a selection in the rows and columns of the various matrices in Eq. (4-3). This can be done with i_{sp}^x, which is a list with all the non-zero indices of $\mathbf{\Phi}_x$ compared to its dense form $\mathbf{\Phi}_x^{\text{dense}}$. Its counterpart for $\mathbf{\Phi}_u$ is denoted by i_{sp}^u. Thus, if $\mathbf{\Phi}_x$ has the following structure with the indices of $\mathbf{\Phi}_x^{\text{dense}}$ labelled as follows:

$$\mathbf{\Phi}_x^{\text{dense}} = \begin{bmatrix} 0 & 3 & 6 \\ 1 & 4 & 7 \\ 2 & 5 & 8 \end{bmatrix}, \quad \mathbf{\Phi}_x = \begin{bmatrix} * & & * \\ & * & \\ * & * & \end{bmatrix},$$

then i_{sp}^x is defined as $[0, 2, 4, 5, 6]$.

Sparse System-Level Synthesis

The optimisation problem is shown in Eq. (4-4), with these sparsity indices used to select the rows and columns corresponding to the sparsity structure:

$$\begin{aligned} \underset{\boldsymbol{\varphi}_x^0, \boldsymbol{\varphi}_u^0}{\text{minimize}} \quad & {\boldsymbol{\varphi}_x^0}^{\mathsf{T}} \mathcal{Q}^{\varphi}[i_{\text{sp}}^x, i_{\text{sp}}^x]\, \boldsymbol{\varphi}_x^0 + {\boldsymbol{\varphi}_u^0}^{\mathsf{T}} \mathcal{R}^{\varphi}[i_{\text{sp}}^u, i_{\text{sp}}^u]\, \boldsymbol{\varphi}_u^0 \\ \text{subject to} \quad & \begin{bmatrix} (I - Z\mathcal{A}^{\varphi})[i_{\text{sp}}^x, i_{\text{sp}}^x] & -(Z\mathcal{B}_2^{\varphi})[i_{\text{sp}}^x, i_{\text{sp}}^u] \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi}_x^0 \\ \boldsymbol{\varphi}_u^0 \end{bmatrix} = \mathcal{I}^{\varphi}[i_{\text{sp}}^x], \\ & x_{\min} \le \mathcal{K}_x[:, i_{\text{sp}}^x]\boldsymbol{\varphi}_x^0 \le x_{\max}, \\ & u_{\min} \le \mathcal{K}_u[:, i_{\text{sp}}^u]\boldsymbol{\varphi}_u^0 \le u_{\max}. \end{aligned} \tag{4-4}$$

4-2 Robust System-Level Synthesis

With the work from the previous sections, it should now be possible to optimise the robust SLS problem significantly faster. However, the main bottleneck remains the nonlinear functions that appear in the constraints of a lumped SLS formulation. Therefore, an equivalent problem is constructed with a lower computational load.

First, the SLP is rewritten in Section 4-2-1, after which the lumped uncertainty is bounded in Section 4-2-2 and the constraints are tightened accordingly in Section 4-2-3. Finally, the resulting optimisation problem is shown in Section 4-2-4.

4-2-1 Robust Dynamic Constraints

The sparse form of the transformed, robust SLP is derived in Section E-2-2, and is equal to:

$$\begin{bmatrix}(I - Z\mathcal{A}^{\varphi})[i^x_{\mathrm{sp},t}, i^x_{\mathrm{sp},t}] & -\mathcal{B}^{\varphi}_2[i^x_{\mathrm{sp},t}, i^u_{\mathrm{sp},t}]\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{\varphi}}^t_x \\ \tilde{\boldsymbol{\varphi}}^t_u\end{bmatrix} = \mathcal{I}^{\varphi}_t[i^x_{\mathrm{sp},t}], \quad \forall t \in \mathbb{Z}^T_0, \tag{4-5}$$

where $\tilde{\boldsymbol{\varphi}}^t_x$, $\tilde{\boldsymbol{\varphi}}^t_u$ and $\mathcal{I}^{\varphi}_t$ are defined as:

$$\tilde{\boldsymbol{\varphi}}^t_x := \begin{bmatrix}\tilde{\varphi}^{t+1,1}_x \\ \vdots \\ \tilde{\varphi}^{T,T-t}_x\end{bmatrix}, \quad \tilde{\boldsymbol{\varphi}}^t_u := \begin{bmatrix}\tilde{\varphi}^{t,0}_u \\ \vdots \\ \tilde{\varphi}^{T-1,T-t-1}_u\end{bmatrix}, \quad \mathcal{I}^{\varphi}_0 := A^{\varphi}\begin{bmatrix}I^{\mathrm{flat}}_{n_x} \\ 0 \\ \vdots \\ 0\end{bmatrix}, \quad \mathcal{I}^{\varphi}_t := A^{\varphi}\begin{bmatrix}\Sigma^{\mathrm{flat}}_{t-1} \\ 0 \\ \vdots \\ 0\end{bmatrix}, \ \forall t \in \mathbb{Z}^T_1.$$

Here, $\tilde{\boldsymbol{\varphi}}^t_x$ and $\tilde{\boldsymbol{\varphi}}^t_u$ represent the t'th block column of $\boldsymbol{\Phi}_x$ and $\boldsymbol{\Phi}_u$, respectively. Note that the indices for the entries of $\tilde{\boldsymbol{\varphi}}^t_x$ are one higher than before, as the values for $\tilde{\Phi}^{t,0}_x$ are known beforehand (i.e., Σ_{t-1}) and have been moved into $\mathcal{I}^{\varphi}_t$ as Σ_{t-1}.

4-2-2 Lumped Uncertainty Bounds

The lumped uncertainty bounds are given in Eqs. (3-20) and (3-21):

$$\begin{aligned}
\left|\eta^j_0\right| &\leq \epsilon^j_A \left\|\tilde{\Phi}^{0,0}_x x_0\right\|_\infty + \epsilon^j_{B_2}\left\|\tilde{\Phi}^{0,0}_u x_0\right\|_\infty + \sigma^j_\delta \leq \sigma^j_0, \quad \forall j \in \mathbb{Z}^{n_x-1}_0, \\
|\eta^j_t| &\leq \epsilon^j_A\left(\left\|\tilde{\Phi}^{t,t}_x x_0\right\|_\infty + \sum_{i=1}^{t}\left\|\tilde{\Phi}^{t,t-i}_x\right\|_{\infty\to\infty}\right) \\
&\quad + \epsilon^j_{B_2}\left(\left\|\tilde{\Phi}^{t,t}_u x_0\right\|_\infty + \sum_{i=1}^{t}\left\|\tilde{\Phi}^{t,t-i}_u\right\|_{\infty\to\infty}\right) + \sigma^j_\delta \leq \sigma^j_t, \quad \forall j \in \mathbb{Z}^{n_x-1}_0, \quad \forall t \in \mathbb{Z}^{T-1}_1,
\end{aligned}$$

and can, as shown in Section E-2-3, equivalently be represented as:

$$\begin{aligned}
\epsilon^j_A x^{\max}_0 + e^j_{B_2} u^{\max}_0 + \sigma^j_\delta &\leq \sigma^j_0, \quad \forall j \in \mathbb{Z}^{n_x-1}_0, \\
\epsilon^j_A\left(x^{\max}_1 + \sigma^{\max}_0\right) + e^j_{B_2}\left(u^{\max}_1 + \tilde{\varphi}^{1,0}_{u,\max}\right) + \sigma^j_\delta &\leq \sigma^j_1, \quad \forall j \in \mathbb{Z}^{n_x-1}_0, \\
\epsilon^j_A\left(x^{\max}_t + \sum_{i=1}^{t-1}\tilde{\varphi}^{t,t-i}_{x,\max} + \sigma^{\max}_{t-1}\right)& \\
+ \epsilon^j_{B_2}\left(u^{\max}_t + \sum_{i=1}^{t}\tilde{\varphi}^{t,t-i}_{u,\max}\right) + \sigma^j_\delta &\leq \sigma^j_t, \quad \forall j \in \mathbb{Z}^{n_x-1}_0, \quad \forall t \in \mathbb{Z}^{T-1}_2.
\end{aligned} \tag{4-6}$$

The dummy variables, all with either the subscript or superscript 'max', must be minimised and require several constraints for this alternative representation to hold. For $x^{\max}_t$ and $u^{\max}_t$, these constraints have the following form:

$$\begin{aligned} x_{t,i}^{+} + x_{t,i}^{-} &\leq x_t^{\max} \quad \forall i \in \mathbb{Z}_0^{n_x-1}, \\ x_t^{+} - x_t^{-} &= K_x[:, i_{\text{sp},t}^x]\tilde{\varphi}^{t,t}, \\ x_t^{+} \geq 0,&\ \ x_t^{-} \geq 0. \end{aligned}$$

For $\tilde{\varphi}_{x,\max}^{t,t-i}$ and $\tilde{\varphi}_{u,\max}^{t,t-i}$, these take the form of:

$$\begin{aligned} N_x^j[:, i_{\text{sp},t}^x](\tilde{\varphi}_x^{t,t-i,+} + \tilde{\varphi}_x^{t,t-i,-}) &\leq \tilde{\varphi}_{x,\max}^{t,t-i} \quad \forall j \in \mathbb{Z}_0^{n_x-1}, \\ \tilde{\varphi}_x^{t,t-i,+} - \tilde{\varphi}_x^{t,t-i,-} &= \tilde{\varphi}_x^{t,t-i}, \qquad (4\text{-}7) \\ \tilde{\varphi}_x^{t,t-i,+} \geq 0,&\ \ \tilde{\varphi}_x^{t,t-i,-} \geq 0, \qquad (4\text{-}8) \end{aligned}$$

where N_x^j is defined as:

$$N_x^j := \mathbf{1}_{n_x}^{\mathsf{T}} \otimes e_j^{\mathsf{T}}.$$

The maximum uncertainty $\sigma_t^{\max}$ simply requires:

$$\sigma_{t-1}^i \leq \sigma_{t-1}^{\max}, \quad \forall i \in \mathbb{Z}_0^{n_x-1}.$$

4-2-3 Constraint Tightening

The tightened constraints are shown in Eq. (3-22), and denoted below once more for convenience:

$$\begin{aligned} H_{\mathcal{X}_t}^j \tilde{\Phi}_x^{t,t} x_0 + \sum_{i=1}^{t-1} \left\| H_{\mathcal{X}_t}^j \tilde{\Phi}_x^{t,t-i} \right\|_1 + \left\| H_{\mathcal{X}_t}^j \Sigma_{t-1} \right\|_1 &\leq h_{\mathcal{X}_t}^j, \ \forall j \in \mathbb{Z}_0^{n\mathcal{X}_t-1}, \ \forall t \in \mathbb{Z}_1^{T-1}, \\ H_{\mathcal{X}_T}^j \tilde{\Phi}_x^{T,T} x_0 + \sum_{i=1}^{T-1} \left\| H_{\mathcal{X}_T}^j \tilde{\Phi}_x^{T,T-i} \right\|_1 + \left\| H_{\mathcal{X}_T}^j \Sigma_{T-1} \right\|_1 &\leq h_{\mathcal{X}_T}^j, \ \forall j \in \mathbb{Z}_0^{n\mathcal{X}_T}, \\ H_{\mathcal{U}_t}^j \tilde{\Phi}_u^{t,t} x_0 + \sum_{i=1}^{t} \left\| H_{\mathcal{U}_t}^j \tilde{\Phi}_u^{t,t-i} \right\|_1 &\leq h_{\mathcal{U}_t}^j, \ \forall j \in \mathbb{Z}_0^{n\mathcal{U}_t-1}, \ \forall t \in \mathbb{Z}_0^{T-1}, \end{aligned}$$

where the one-norms must be replaced to solve the problem using a standard quadratic solver. As shown in Section E-2-4, these constraints can be replaced by:

$$\begin{aligned} & H_{\mathcal{X}_t}^j K_x[:, i_{\text{sp},t}^x]\tilde{\varphi}_x^{t,t} \\ & \qquad + \sum_{i=1}^{t-1} N_x^{m_j^x}[:, i_{\text{sp},t}^x](\tilde{\varphi}_x^{t,t-i,+} + \tilde{\varphi}_x^{t,t-i,-}) + \sigma_{t-1}^{m_j^x} \leq h_{\mathcal{X}_t}^j, \ \forall j \in \mathbb{Z}_0^{n\mathcal{X}_t-1}, \ \ \forall t \in \mathbb{Z}_1^{T-1}, \\ & H_{\mathcal{X}_T}^j K_x[:, i_{\text{sp},t}^x]\tilde{\varphi}_x^{T,T} \qquad (4\text{-}9) \\ & \qquad + \sum_{i=1}^{T-1} N_x^{m_j^x}[:, i_{\text{sp},t}^x](\tilde{\varphi}_x^{T,T-i,+} + \tilde{\varphi}_x^{T,T-i,-}) + \sigma_{T-1}^{m_j^x} \leq h_{\mathcal{X}_T}^j, \ \forall j \in \mathbb{Z}_0^{n\mathcal{X}_T-1}, \\ & H_{\mathcal{U}_t}^j K_u[:, i_{\text{sp},t}^u]\tilde{\varphi}_u^{t,t} + \sum_{i=1}^{t} N_u^{m_j^u}[:, i_{\text{sp},t}^u](\tilde{\varphi}_u^{t,t-i,+} + \tilde{\varphi}_u^{t,t-i,-}) \leq h_{\mathcal{U}_t}^j, \ \forall j \in \mathbb{Z}_0^{n\mathcal{U}_t-1}, \ \ \forall t \in \mathbb{Z}_0^{T-1}, \end{aligned}$$

where m_j^x and m_j^u are defined with the modulo operation as:

$$m_j^x := j \bmod n_x,$$
$$m_j^u := j \bmod n_u,$$

and under the following three conditions:

1. The dummy variables (i.e., terms with the superscripts $^+$ or $^-$) are minimised.
2. The same constraints for these dummy variables as in Eqs. (4-7) and (4-8) are added to the optimisation problem.
3. The state (and similarly for the input constraints) are of the form $x_{\text{min}} \leq x_t \leq x_{\text{max}}$.

4-2-4 Robust System-Level Synthesis Problem

It is possible to combine the results from the previous sections as follows:

$$\begin{aligned} \underset{\boldsymbol{\varphi}_x^t, \boldsymbol{\varphi}_u^t, \boldsymbol{\Sigma}}{\text{minimize}} \quad & {\boldsymbol{\varphi}_x^0}^\top \mathcal{Q}^\varphi[i_{\text{sp}}^x, i_{\text{sp}}^x]\, \boldsymbol{\varphi}_x^0 + {\boldsymbol{\varphi}_u^0}^\top \mathcal{R}^\varphi[i_{\text{sp}}^u, i_{\text{sp}}^u]\, \boldsymbol{\varphi}_u^0 \\ \text{subject to} \quad & \text{Dynamical constraints from Eq. (4-5)}, \\ & \text{Lumped uncertainty bounds from Eq. (4-6)}, \\ & \text{Tightened constraints from Eq. (4-9)}. \end{aligned} \tag{4-10}$$

Where previously there were terms representing the one-norm, the infinity-norm or the upper bound on the lumped disturbance in the objective function, these are now absent. These can safely be left out because:

- By not actively minimising these variables, a larger value for these norms or disturbance estimations may be obtained. This represents a larger uncertainty in the model through Eq. (4-6).
- However, if this extra uncertainty forces the controller to follow a different path, this must come at the cost of a larger objective value. If this would not be the case, the controller would have taken that path with less uncertainty before.
- Therefore, the optimisation problem will actively minimise the norms and estimated disturbances to achieve the best possible path unless these norms and disturbances do not result in a larger objective value.

To show this is indeed correct, Fig. 4-1 shows the results for the exact formulation used before in Eq. (3-23) and the quadratic program formulation from Eq. (4-10).

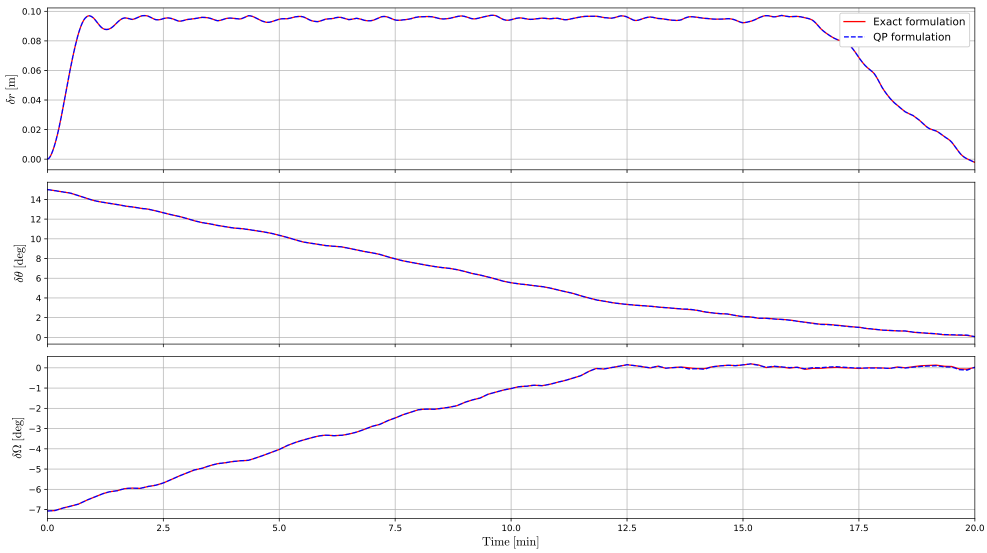


Figure 4-1: Comparison of exact and quadratic variant of lumped System-Level Synthesis.

4-3 Problem Formulation And Solver Comparison

These different problem formulations are tested for various solvers to find the combination that allows for the largest constellation. To do this, several solvers are discussed in Section 4-3-1 and afterwards an initial selection is made with a regular MPC problem in Section 4-3-2. The most promising solvers are then tested on the nominal SLS problems in Section 4-3-3 and the robust SLS problem in Section 4-3-4.

4-3-1 Solver Selection

There are many solvers available to find the optimal solution to this problem, of which the following five are selected to test their performance:

- `quadprog`. This is the standard solver for quadratic problems in MATLAB, but versions exist as well for other languages such as Python and R. `quadprog` can be used both with sparse and dense matrices, which makes it a suitable choice to show how that can affect the results.
- `Gurobi`. This commercial solver (with academic licenses) can solve a large set of optimisation problems and is generally one of the fastest solvers available. The performance and its applicability to many different types of problems leads companies such as Google, Amazon and Apple to use `Gurobi`. This solver is added to compare the results with the industry standard.
- `OSQP` [37]. This is a state-of-the-art solver for quadratic problems and has quickly gained popularity with interfaces in Python, MATLAB, C, R and Julia. This solver has outperformed `Gurobi` for certain problems. It is added to give an insight into the state-of-the-art performance of quadratic problem solvers.

- `cuOSQP` [38]. This GPU-variant of `OSQP` has shown improved performance for large problems. It is the only solver in this list that utilises a GPU and one of the few that does so in general, as most optimisation problems are still solved using CPUs. This solver provides insight into the growing group of GPU solvers.

- `CVX`. Although `CVX` is merely a problem parser and it uses other solvers (such as `Gurobi` and `OSQP`), it is included in the comparison as it is commonly used in the academic world. For example, both the SLS toolboxes in MATLAB and Python use `CVX` to find their answer. Its use in the comparison is two-fold: it provides an overview of the loss in speed when using such as sparser and also gives an overview into the speed of the toolboxes.

These solvers are tested on problems with Blend model and the parameters from Section B-1-3. Converging parameters differ for each solver but are set such that the absolute and relative tolerance is 10^{-3}. All simulations are run with an Intel i7-7700HQ CPU and, where appropriate, an NVidia GeForce RTX 3090 GPU.

4-3-2 Standard Model Predictive Control Problems

Although eventually the robust SLS algorithm is used to control the satellites, the first comparison between the solvers is with a basic MPC formulation. This MPC problem is added for two reasons:

1. It provides a first overview of the different solvers, such that solvers that are significantly slower can be discarded early.

2. It allows for a comparison in terms of speed between a nominal SLS formulation and a MPC formulation.

An MPC problem is often a quadratic problem of the following form:

$$\begin{aligned} \underset{\substack{x_0,\ldots,x_T,\\ u_0,\ldots,u_{T-1}}}{\text{minimize}} \quad & \sum_{t=0}^{T-1}\left[x_t^{\mathsf{T}}\,Q\,x_t + u_t^{\mathsf{T}}\,R\,u_t\right] + x_T^{\mathsf{T}}\,Q_T\,x_T \\ \text{subject to} \quad & x_{t+1} = A_t\,x_t + B_t\,u_t, \quad \forall t\in\mathbb{Z}_0^{T-1}, \\ & x_{\min}\le x_t\le x_{\max}, \quad \forall t\in\mathbb{Z}_1^{T}, \\ & u_{\min}\le u_t\le u_{\max}, \quad \forall t\in\mathbb{Z}_0^{T-1}. \end{aligned} \tag{4-11}$$

Note how this MPC formulation controls the state variables x_t to zero, as no explicit reference value is given. For linear systems, it is possible to find an equivalent formulation where this reference is explicitly provided:

$$\begin{aligned} \underset{\substack{x_0,\ldots,x_T,\\ u_0,\ldots,u_{T-1}}}{\text{minimize}} \quad & \sum_{t=0}^{T-1}\left[(x_t-x_r)^{\mathsf{T}}\,Q\,(x_t-x_r) + u_t^{\mathsf{T}}\,R\,u_t\right] + (x_T-x_r)^{\mathsf{T}}\,Q_T\,(x_T-x_r) \\ \text{subject to} \quad & x_{t+1} = A_t\,x_t + B_t\,u_t, \quad \forall t\in\mathbb{Z}_0^{T-1}, \\ & x_{\min}\le x_t\le x_{\max}, \quad \forall t\in\mathbb{Z}_1^{T}, \\ & u_{\min}\le u_t\le u_{\max}, \quad \forall t\in\mathbb{Z}_0^{T-1}. \end{aligned} \tag{4-12}$$

For the OSQP and cuOSQP solvers, it is found that solving Eq. (4-12) increased the convergence rate of the solver significantly compared to Eq. (4-11).

The average computation time to solve Eq. (4-11) or Eq. (4-12) for the various solvers are shown in Fig. 4-2. There are a few important observations that can be drawn from these results:

- The quadprog solver, which is the only dense solver, scales with $\mathcal{O}(n^3)$. With a computation time of one second for six satellites, it would take approximately 11.5 hours to find the solution for 600 satellites.
- The bulk of the solvers, namely Gurobi, OSQP and CVX, scale with $\mathcal{O}(n)$. OSQP outperforms Gurobi in terms of speed both through CVX and by using them directly. The results for Gurobi are more predictable due to the lower variance.
- The results for cuOSQP show different behaviour compared to the CPU solvers, especially for the smaller problems. For the larger problems, the results scale once more with $\mathcal{O}(n)$. The odd behaviour for smaller problems is likely caused by the GPU's inability to use its parallel abilities fully. A similar result can be found in [38].

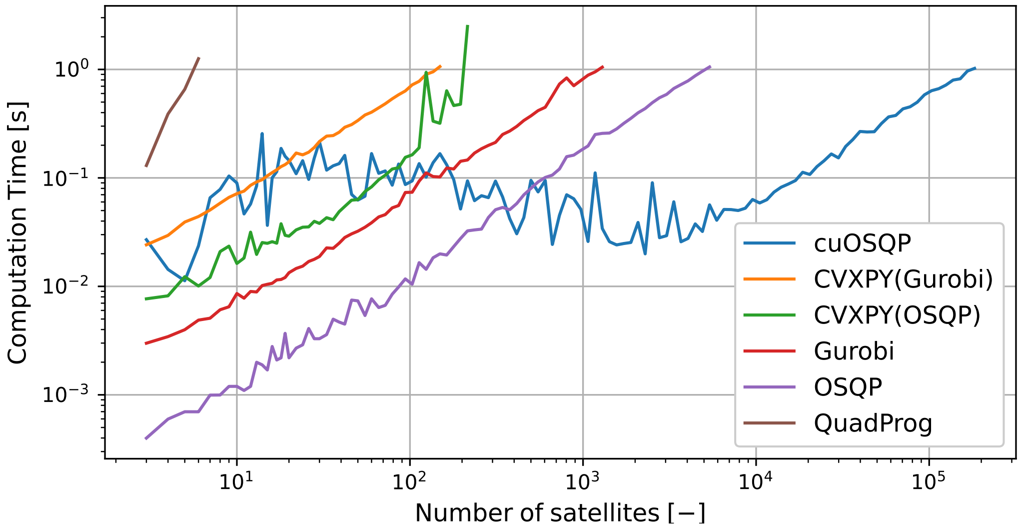


Figure 4-2: Required solver time for a basic Model Predictive Control problem.

As the quadprog solver scales poorly and is already the slowest solver, it is not used for the SLS problems. Furthermore, because the CVX parser is ten times slower than when using the solvers directly, that solver too is not used for the SLS problems.

4-3-3 Nominal System-Level Synthesis Problems

Three formulations for the nominal SLS problem have been presented in Section 4-1. A comparison between these three different formulations is provided first, after which the Gurobi, OSQP and cuOSQP solvers are compared to each other and a benchmark solver to find the best nominal SLS solver.

Problem Formulation Comparison

Three nominal SLS problem formulations are discussed in Section 4-1: the default, dense variant with matrix decision variables, the dense vector variant and its sparse counterpart. To show the effect of these changes, `Gurobi` is used to solve all three problem formulations as this is the only solver[2] that can handle matrix variables.

The results with `Gurobi` are shown in Fig. 4-3. The original, dense matrix formulation requires the most computation time. Transforming the problem such that it uses vectors as decision variables approximately halves the required computation time. The computation time of both problems scales with the size of the matrices $\mathbf{\Phi}_x^0$ and $\mathbf{\Phi}_u^0$, thus with $\mathcal{O}(n^2)$.

The sparse variant is significantly faster. For small problems, the sparse problem formulation is roughly five times faster than the transformed formulation and ten times faster than the original formulation. However, as the number of decision variables now scales linearly with the number of satellites, the computation time of the sparse problem formulation scales with $\mathcal{O}(n)$ and the difference between the dense and sparse formulation grows for larger problems.

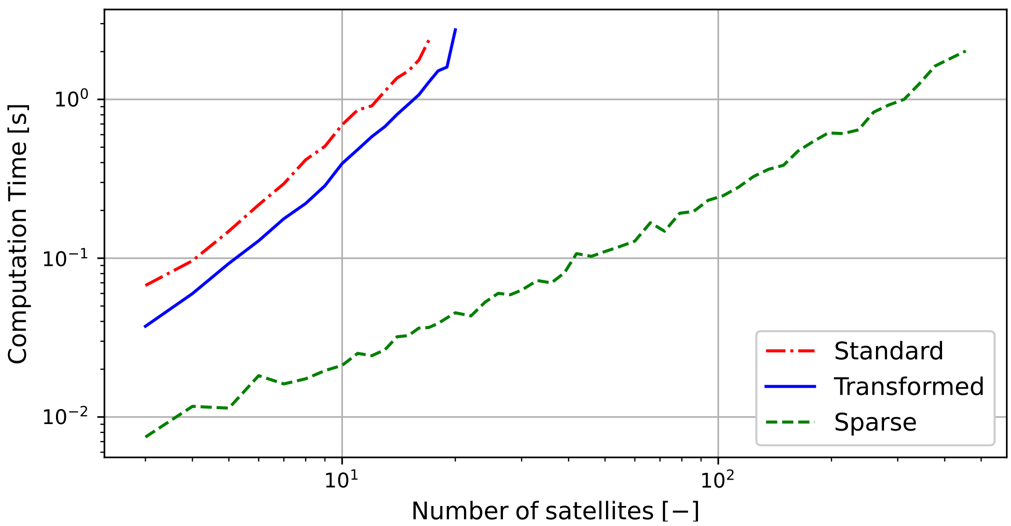


Figure 4-3: Computation time for different nominal problem formulations with `Gurobi`.

Table 4-1 shows the required computation time for four different satellite constellations. All times denoted with * were extrapolated, and those values should be considered as an approximation. However, they give a clear indication of the scale at larger constellations, where the sparse SLS variant can solve problems in seconds that take the other two variants hours.

Table 4-1: Computation time for various constellation sizes with `Gurobi`.

Formulation	3 satellites	10 satellites	100 satellites	1000 satellites
Standard	0.0673 s	0.6920 s	78.40* s	9184* s
Transformed	0.0373 s	0.3945 s	58.55* s	8393* s
Sparse	0.0075 s	0.0211 s	0.2422 s	3.383* s

[2]The only solver out of the remaining three: `Gurobi`, `OSQP` and `cuOSQP`.

Solver Comparison

The fastest solvers for a simple MPC problem are, as shown in Section 4-3-2, the `OSQP` and `cuOSQP` solvers. The sparse formulation that proved to be significantly faster than the standard SLS formulation is implemented for these two solvers, and compared with a Python toolbox for SLS [39] and with `Gurobi`.

The results are shown in Fig. 4-4, where the `OSQP` solver outperforms the other solvers up to one thousand satellites. The `cuOSQP` solver is relatively slow for smaller problems but becomes increasingly effective when the problem size grows. The toolbox, used as a benchmark, is orders of magnitude slower, especially when the number of satellites increases as it scales with $\mathcal{O}(n^2)$, similarly to the standard formulation as seen in Fig. 4-3.

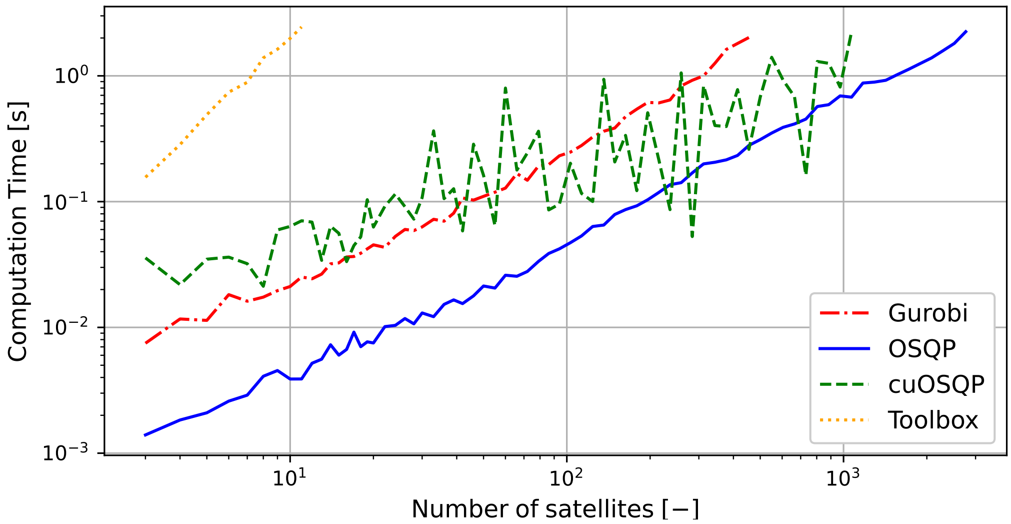


Figure 4-4: Computation time for different solvers with the sparse problem formulation.

Table 4-2 provides several quantitative values to compare these solvers, where similarly to Table 4-1 the * denotes an extrapolated value. The `OSQP` solver is consistently approximately five times faster than `Gurobi` and consistently faster than `cuOSQP`, although that margin decreases. For even larger problems, it is possible that `cuOSQP` outperforms `OSQP`, similarly to what is seen in Fig. 4-2. The toolbox is approximately three times slower than the standard SLS implementation in `Gurobi` from Table 4-1, and it is estimated that it would take the toolbox nearly ten hours to find the solution for a thousand satellites.

Table 4-2: Computation time for various constellation sizes and various solvers.

Formulation	3 satellites	10 satellites	100 satellites	1000 satellites
`Gurobi`	0.0075 s	0.0211 s	0.2422 s	3.383* s
`OSQP`	0.0014 s	0.0039 s	0.0455 s	0.6882 s
`cuOSQP`	0.0357 s	0.0635 s	0.1586 s	1.1018 s
Toolbox	0.1562 s	1.985 s	270.0* s	35606* s

4-3-4 Robust System-Level Synthesis Problems

The robust SLS problem from Eq. (4-10) is implemented in `Gurobi`, `OSQP` and `cuOSQP`. Where `OSQP` performed well for the previous problem, it was unable to converge for this problem with sufficient accuracy to be comparable to the other solvers. Therefore, no results for `OSQP` are visible in Fig. 4-5. The four solvers that are visible are:

1. The `Gurobi` implementation of Eq. (4-10) named `Gurobi QP`.
2. The `Gurobi` implementation of the nonlinear robust ILSLS problem from Eq. (3-23). This version is named `Gurobi NL`.
3. The `cuOSQP` version of Eq. (4-10) named `cuOSQP`.
4. The CLSLS implementation from [28] denoted by `CLSLS`. This is a numerically slightly simpler problem than the ILSLS problems the other solvers try to solve, as there are fewer decision variables[3].

The results are shown in Fig. 4-5, where the `Gurobi QP` version outperforms all other solvers. The nonlinear variant `Gurobi NL` is on average approximately ten times faster than the `CLSLS` problem. The GPU implementation of `cuOSQP` requires larger problems before it can fully make use of its parallel computations. For larger problems, however, similar problems to the `OSQP` solver are encountered, and no reliable results are obtained. Therefore, these results are limited to fourteen satellites.

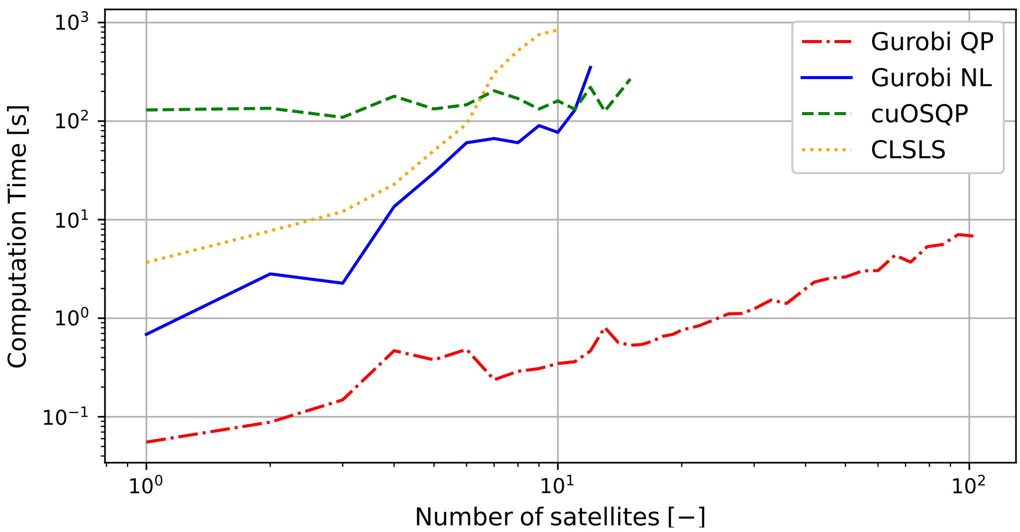


Figure 4-5: Computation time for various solvers for a robust problem formulation.

Table 4-3 shows the obtained and estimated (denoted with *) computation times for the four solvers. `Gurobi QP` is estimated to be five thousand times faster for 100 satellites than the nonlinear variant `Gurobi NL` and outperforms the `CLSLS` solver with an even greater margin. `cuOSQP` is not extended to 100 satellites as the solver was unable to produce accurate results for larger problems.

[3]See Section 3-3 for a more detailed explanation.

Table 4-3: Computation time for various solvers for a robust problem.

Formulation	1 satellite	10 satellites	100 satellites
`Gurobi QP`	0.0554 s	0.3468 s	6.879 s
`Gurobi NL`	0.6862 s	76.89 s	32024* s
`cuOSQP`	129.7 s	160.4 s	-
`CLSLS`	3.6848 s	839.7 s	220360* s

4-4 Conclusions

To scale the problem to hundreds or thousands of satellites, it is important to select an appropriate solver and formulate the problem accordingly. For a standard MPC problem, state-of-the-art solvers such as `OSQP` and `cuOSQP` are shown to be faster than commercial solvers such as `Gurobi` or their parsed versions from `CVXPY`. These state-of-the-art solvers can find the optimal solution for thousands (`OSQP`) to hundred thousands (`cuOSQP`) of satellites within a second.

However, this relatively basic formulation does not account for model uncertainties or disturbances. When including those with the SLS framework, three steps are taken to speed up the computations:

- The model is transformed from a matrix problem to a standard quadratic problem.
- A sparse optimisation problem is constructed, where optimisation variables that must be zero are removed.
- Nonlinear elements such as the infinity and one-norms are rewritten to obtain a quadratic problem.

By combining all these elements, the optimisation problem can outperform the available toolbox significantly by using `Gurobi`. For a problem of 100 satellites, this new approach can solve the problem in seconds where it takes the toolbox days.

With the scale of the problems increasing, it is important to check for collisions between satellites. This is discussed in the next chapter.

Chapter 5

Large-Scale Simulation

The previous chapters have discussed a novel dynamical model, a novel control algorithm and an optimisation formulation to reduce the computation time for the solvers. This is all done to answer the research question, where the goal is to control a large-scale constellation robustly. These different results are combined here to show that it is indeed possible to do so.

First, the assignment problem is discussed in Section 5-1, which is followed by the collision avoidance constraints that are added in Section 5-2. The results are shown in Section 5-3, followed by the respective conclusions in Section 5-4.

5-1 Assignment Problem

The scenario of the large simulation is as follows:

- Before the start of the simulation, there were a total of 240 satellites divided over fifteen evenly-spaced planes, equalling sixteen satellites per plane.
- Fifteen of these satellites are randomly selected and drop out. This can be because of maintenance, collisions or because these satellites are now used in another constellation.
- The remaining 225 satellites have to reconfigure themselves such that fifteen satellites are spaced out around the orbit in each plane.

The remaining satellites can, once they have a target plane and argument of latitude, move to this position using the dynamical model and control method as discussed in Chapters 2 and 3. However, finding the optimal target position for each satellite constitutes an assignment problem, where given the start locations, a set of target locations must be reached.

The assignment problem is solved using the so-called Hungarian or (Kuhn-)Munkres algorithm, as this algorithm can find the optimal solution in polynomial time. The algorithm is optimal in the sense that it minimises a given objective function, where the cost represents the distance from the starting to the final position. The objective function J_{Hung} for the Hungarian algorithm is defined as:

$$J_{\text{Hung}}(\theta^0, \Omega^0, \theta^f, \Omega^f) := f_{\text{dist}}(\theta^f - \theta^0)^2 + f_{\text{dist}}(\Omega^f - \Omega^0)^2 + f_{\text{dir}}(\theta^f - \theta^0, \Omega^f - \Omega^0),$$

where the superscripts $(\cdot)^0$ and $(\cdot)^f$ denote the values at the start and end respectively, where $f_{\text{dist}}(x)$ denotes the distance on a circle:

$$f_{\text{dist}}(x) := \min\left(|x|, 2\pi - |x|\right),$$

and where $f_{\text{dir}}(x, y)$ provides an extra penalty if x and y have the same sign while both being expressed in the range $[-\pi, \pi]$:

$$f_{\text{dir}}(x, y) = \begin{cases} 0, & \text{if } x \cdot y \leq 0, \\ x \cdot y, & \text{if } x \cdot y > 0. \end{cases}$$

The reasoning for this objective function is threefold:

- The basis of the objective function is provided through f_{dist}, which minimises the required distance to travel.
- The function f_{dist} is squared to encourage smaller movements. Rather than a single satellite moving twenty degrees between different planes, two satellites moving ten degrees each would be preferred as this is faster and spreads the (fuel) load.
- The function f_{dir} is added to help with the in-plane collision constraints discussed in Section 5-2-1. This function favours trajectories that result in a smaller initial $\delta\lambda^f$, decreasing the relative changes in λ^f between satellites.

The result obtained with the Hungarian algorithm and this cost function is shown in Fig. 5-1, where all satellites move to their desired position while satisfying the desired behaviour as explained above.

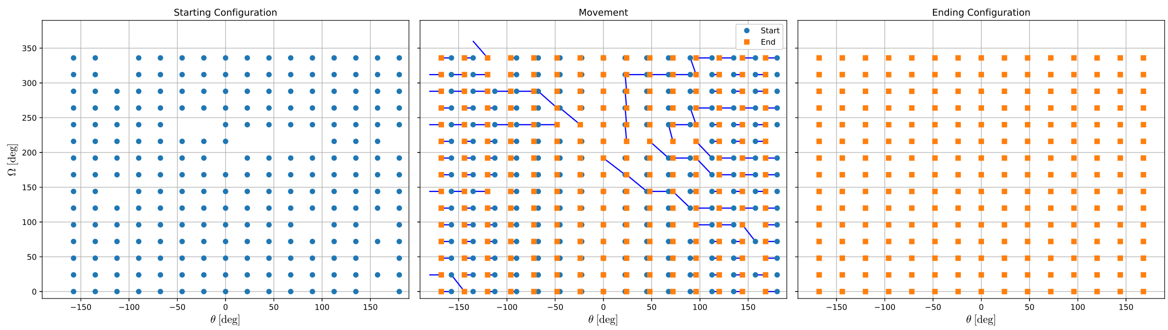


Figure 5-1: Result of assignment problem with the Hungarian algorithm.

5-2 Collision Avoidance Constraints

The collision avoidance constraints are divided into two different parts. The first part, discussed in Section 5-2-1, discusses the constraints to prevent collisions between satellites within the same plane. The second part, presented in Section 5-2-2, deals with preventing collisions between satellites in different planes.

5-2-1 In-Plane Collision Avoidance Constraints

The in-plane collision avoidance constraint primarily aims to prevent collisions between satellites in the same plane. This means preventing two different satellites in the same plane (i.e., the same Ω) from having the same argument of latitude (i.e., the same θ). Although these parameters are not separately part of the state of the Blend model (see Section 2-2-3), they are present in the variable $\delta\lambda^f$, which as a reminder, is defined as follows for satellite k with a chief (or reference) satellite c:

$$\begin{aligned}\delta\lambda_k^f &= \theta_{d,k} - \theta_{c,k} + \cos i_{c,k}(\Omega_{d,k} - \Omega_{c,k}) \\ &= \theta_{d,k} + \cos i_{c,k}\Omega_{d,k} - (\theta_{c,k} + \cos i_{c,k}\Omega_{c,k}) \\ &= \lambda_k^f - \lambda_{\text{ref},k}^f,\end{aligned} \tag{5-1}$$

with λ_k^f and $\lambda_{\text{ref},k}^f$ defined as:

$$\begin{aligned}\lambda_k^f &:= \theta_{d,k} + \cos i_{c,k}\Omega_{d,k}, \\ \lambda_{\text{ref},k}^f &:= \theta_{c,k} + \cos i_{c,k}\Omega_{c,k}.\end{aligned}$$

Satellites that collide because they are in the same plane with the same argument of latitude have the same λ^f. Therefore, a simple constraint can be constructed to prevent a collision between two satellites m and n:

$$\lambda_m^f - \lambda_n^f \geq \lambda_{\text{min}}^f, \tag{5-2}$$

where λ_{min}^f is the minimal difference between two satellites. This must always be a positive number to prevent any collisions, but it can also deliberately be set relatively high to force the satellites to space out during the reconfiguration and, therefore, keep better coverage of the Earth.

Note that a violation of this constraint does not guarantee a collision, i.e. two satellites with the same λ^f do not necessarily collide as they can be in different planes. The conservativeness of this constraint does not have to be a problem, as is demonstrated in Section 5-3.

Nominal and Robust Constraint

Before that can be done, however, the constraint has to be written into a formulation usable for both the nominal and the robust controller. For the nominal case, this can be done by simply combining Eqs. (5-1) and (5-2):

$$\delta\lambda_m^f - \delta\lambda_n^f \geq \lambda_{\text{min}}^f - \delta\lambda_{\text{ref},m}^f + \delta\lambda_{\text{ref},n}^f, \tag{5-3}$$

where the state variables $\delta\lambda_k^f$ are all on the left-hand side, and all constants are on the right-hand side. For the robust controller, the constraint takes the form from Eq. (3-22):

$$H_{\mathcal{X}_t}^j \tilde{\Phi}_x^{t,t} x_0 + \sum_{i=1}^{t-1} \left\| H_{\mathcal{X}_t}^j \tilde{\Phi}_x^{t,t-i} \right\|_1 + \left\| H_{\mathcal{X}_t}^j \Sigma_{t-1} \right\|_1 \leq h_{\mathcal{X}_t}^j, \ \forall j \in \mathbb{Z}_0^{n_{\mathcal{X}_t}-1}, \ \forall t \in \mathbb{Z}_1^{T-1},$$

which, after accounting for flipping the 'greater than' symbol in Eq. (5-3), has the following $H_{\mathcal{X}_t}$ and $h_{\mathcal{X}_t}$:

$$H_{\mathcal{X}_t} = e_n - e_m, \quad h_{\mathcal{X}_t} = -\lambda_{\text{min}}^f + \delta\lambda_{\text{ref},m}^f - \delta\lambda_{\text{ref},n}^f.$$

Using the same simplification steps as discussed in Section E-2-4, the constraints can be simplified to:

$$(e_{m_n^x} - e_{m_m^x})\tilde{\Phi}_x^{t,t} x_0 + \sum_{i=1}^{t-1} \left(\left\| \tilde{\Phi}_x^{t,t-i,m_m^x} \right\|_1 + \left\| \tilde{\Phi}_x^{t,t-i,m_n^x} \right\|_1 \right) + \sigma_{t-1}^{m_m^x} + \sigma_{t-1}^{m_n^x} \leq -\lambda_{\text{min}}^f + \delta\lambda_{\text{ref},m}^f - \delta\lambda_{\text{ref},n}^f,$$

where the uncertainty of both satellites is now combined. How to simplify the norms is also discussed in Section E-2-4.

Constraint Selection

Finally, a selection has to be made between which satellites this constraint has to be applied. The constraints are applied between all neighbours within the same plane to ensure that collisions are prevented during the entire reconfiguration. This holds both for neighbours at the start and end of the simulation. More mathematically, this means that:

$$\begin{aligned}
(i,j) &\in \mathcal{C}_{\text{start}} \;\mid\; \Omega_j^0 - \Omega_i^0 = 0 \wedge \theta_j^0 - \theta_i^0 = \theta_{\text{sep}}^0, \\
(i,j) &\in \mathcal{C}_{\text{end}} \;\mid\; \Omega_j^f - \Omega_i^f = 0 \wedge \theta_j^f - \theta_i^f = \theta_{\text{sep}}^f, \\
(i,j) &\in \mathcal{C}_{\text{total}} \;\mid\; (i,j) \in \mathcal{C}_{\text{start}} \vee (i,j) \in \mathcal{C}_{\text{end}},
\end{aligned}$$

with $\mathcal{C}_{\text{total}}$ representing the tuples of satellites for which the constraint should hold, $\mathcal{C}_{\text{start}}$ and $\mathcal{C}_{\text{end}}$ the set based on the starting and end positions respectively and θ_{sep} the angular separation between two adjacent satellites.

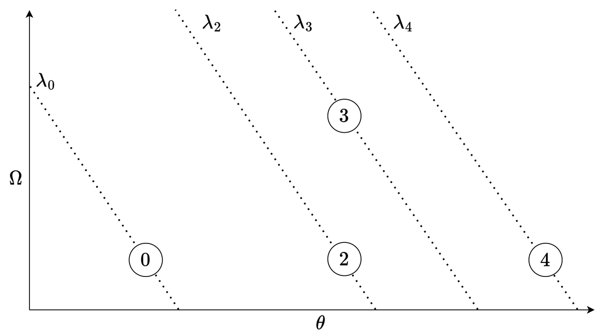
To clarify this further, Fig. 5-2 shows an example of a reconfiguration problem with initially six satellites. Two of these satellites (satellites 1 and 5) have dropped out, and the remaining four satellites have to reconfigure into a formation again. The dashed lines denote the values for which λ_i is constant, where λ_i is the current λ value of satellite i.

For Figs. 5-2a and 5-2b, the following $\mathcal{C}_{\text{start}}$ and $\mathcal{C}_{\text{end}}$ are obtained:

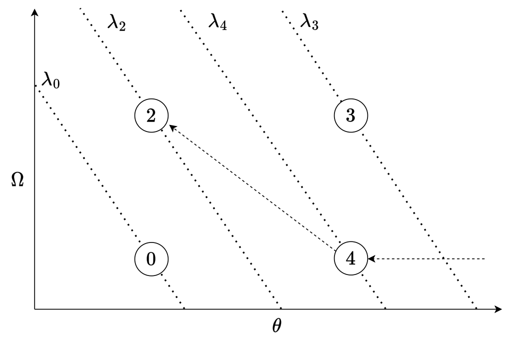
$$\mathcal{C}_{\text{start}} = \{(2,0),(4,2),(0,4)\}, \;\; \mathcal{C}_{\text{end}} = \{(4,0),(0,4),(3,2),(2,3)\},$$

where $(0,4)$ and $(2,3$ are special cases, as the orbit loops back after 2π, and thus these satellites are also adjacent. Furthermore, it should be noted that in a larger scenario, it would never occur that both (i,j) and (j,i) are present in the set. Finally, the combined set then equals:

$$\mathcal{C}_{\text{total}} = \{(2,0),(4,2),(0,4),(3,2),(2,3)\}.$$



(a) Starting position.



(b) Reconfiguration.

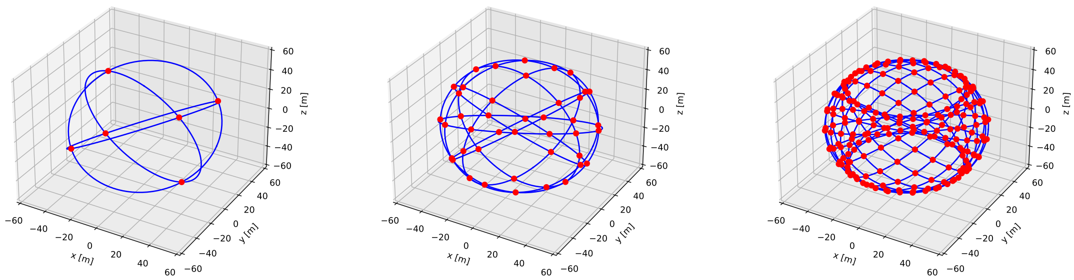
Figure 5-2: Example of in-plane constraints during reconfiguration.

When looking at Fig. 5-2b, it is clear that the constraints between these satellites are never violated, as the contour lines with a constant λ never cross for these satellites. Note that the line for λ_4 crosses the line for λ_3, but as these satellites never share a plane, there is no constraint between these two.

Finally, note the importance of the directional cost for the assignment problem discussed in Section 5-1. Because satellite 2 moves almost along a line of constant λ, the constraint is always satisfied. A similar configuration could be achieved by moving satellite 4 to the empty spot (by looping back through 2π), but in that case, λ_4 crosses with λ_0.

5-2-2 Out-Of-Plane Collision Avoidance Constraints

Where initially the in-plane collision avoidance constraints of Section 5-2-1 might seem enough, the different orbits all cross each other at two points as any two unique circles on a sphere do. With the number of planes rising for a large constellation, so does the number of intersections between these planes. This increases the chance of two satellites in different planes colliding, which is prevented through the out-of-plane collision avoidance constraints. This is visualised in Fig. 5-3, where the number of orbits and the number of collision points, marked in red, increase.



(a) 3 orbits. **(b)** 7 orbits. **(c)** 15 orbits.

Figure 5-3: Example of orbits with collision points.

Firstly, the arguments of latitude for which a collision occurs are derived. This is followed by the general constraint formulation, after which the nominal and robust constraints are presented. Finally, the satellites for which this constraint holds are discussed.

Angle of Collision

To formulate a constraint, it is important to find the respective arguments of latitude for when the satellites would collide first. To find these values, consider two planes with Ω_1 and Ω_2 as the RAAN for each plane and both with inclination i. The unit vectors n_1 and n_2 normal to the orbital planes can easily be found in the ECI frame:

$$n_1 = \begin{bmatrix} \cos\Omega_1 & -\sin\Omega_1 & 0 \\ \sin\Omega_1 & \cos\Omega_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \sin\Omega_1 \sin i \\ -\cos\Omega_1 \sin i \\ \cos i \end{bmatrix},$$

$$n_2 = \begin{bmatrix} \cos\Omega_2 & -\sin\Omega_2 & 0 \\ \sin\Omega_2 & \cos\Omega_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \sin\Omega_2 \sin i \\ -\cos\Omega_2 \sin i \\ \cos i \end{bmatrix},$$

which, when applying the cross product, yields the unit vector n_{coll} along which the planes intersect:

$$n_{\text{coll}} = n_2 \times n_1 = \begin{bmatrix} (\cos\Omega_1 - \cos\Omega_2)\sin i \cos i \\ (\sin\Omega_1 - \sin\Omega_2)\sin i \cos i \\ \sin^2 i \sin(\Omega_1 - \Omega_2) \end{bmatrix}.$$

When expressing this unit vector in the corresponding orbital frames:

$$n^1_{\text{coll}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{bmatrix} \begin{bmatrix} \cos\Omega_1 & \sin\Omega_1 & 0 \\ -\sin\Omega_1 & \cos\Omega_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} n_{\text{coll}} = \begin{bmatrix} (1 - \cos(\Omega_1 - \Omega_2))\sin i \cos i \\ \sin i \sin(\Omega_1 - \Omega_2) \\ 0 \end{bmatrix},$$
$$n^2_{\text{coll}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{bmatrix} \begin{bmatrix} \cos\Omega_2 & \sin\Omega_2 & 0 \\ -\sin\Omega_2 & \cos\Omega_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} n_{\text{coll}} = \begin{bmatrix} (\cos(\Omega_1 - \Omega_2) - 1)\sin i \cos i \\ \sin i \sin(\Omega_1 - \Omega_2) \\ 0 \end{bmatrix},$$

the arguments of latitude can be found accordingly:

$$\theta^1_{\text{coll}} = \arctan\left(\frac{\sin(\Omega_1 - \Omega_2)}{(1 - \cos(\Omega_1 - \Omega_2))\cos i}\right) + k \cdot \pi,$$
$$\theta^2_{\text{coll}} = \arctan\left(\frac{\sin(\Omega_1 - \Omega_2)}{(\cos(\Omega_1 - \Omega_2) - 1)\cos i}\right) + k \cdot \pi.$$

Nominal And Robust Constraint

The constraint has to prevent any two satellites in different planes from reaching θ_{coll} while having the same radius. The radial part is important, as two circular orbits only intersect if they have the same radius. Mathematically, this can be represented as:

$$|r_2 - r_1| + \alpha_{\text{w}}|\theta_2 - \theta_1 - \theta^2_{\text{coll}} + \theta^1_{\text{coll}}| \geq r_{\text{min}}, \tag{5-4}$$

where α_{w} is a constant parameter that weighs the importance of the two terms, and where r_{min} is the minimal radial distance between two satellites at a crossing.

Before it can be applied, it has to be rewritten into a formulation with the state variables. As all satellites have the same radial reference, the first term is simply equal to $|\delta r_2 - \delta r_1|$. The second term can, using Eq. (5-1), be simplified such that the entire constraint is equal to:

$$|\delta r_2 - \delta r_1| + \alpha_{\text{w}}|\delta\lambda^f_2 - \delta\lambda^f_1 + \lambda^f_{\text{ref},2} - \lambda^f_{\text{ref},1} - \cos i(\Omega_2 - \Omega_1) - \theta^2_{\text{coll}} + \theta^1_{\text{coll}}| \geq r_{\text{min}}. \tag{5-5}$$

Two problems remain with this current formulation: the absolute values and the dependence on Ω. Firstly, the absolute values must be rewritten to fit the desired quadratic program formulation. It is possible to use the same approach as in Section 4-2 with dummy variables. However, this can be difficult as these dummy variables need to be minimised while the nature of the constraint is to maximise these dummy variables such that the constraint is met. An extra cost can be added to force this, which can be hard to tune and lead to a suboptimal result.

Furthermore, the required number of constraints quickly grows when increasing the number of planes, meaning a significant number of constraint variables must be added and the computational load increases. Instead, two different approaches are implemented:

- A satellite that decreases its angular velocity and, therefore, decreases its relative λ^f with respect to its reference does so by increasing its radius. Furthermore, the radius shows little to no overshoot in previous simulations when decreasing to zero. Therefore, the sign of $\delta r_2 - \delta r_1$ can be predicted beforehand:

$$r_{\text{sign}}(\lambda_2^{f,0}, \lambda_1^{f,0}) = \begin{cases} 1, & \text{if } \lambda_2^{f,0} \geq \lambda_1^{f,0}, \\ -1, & \text{if } \lambda_2^{f,0} < \lambda_1^{f,0}, \end{cases}$$

  such that the absolute value can be computed as:

$$|\delta r_2 - \delta r_1| = r_{\text{sign}}(\lambda_2^{f,0}, \lambda_1^{f,0}) \cdot (\delta r_2 - \delta r_1).$$

- This approach cannot be followed for the angular term in Eq. (5-5), as during the manoeuvre this sign switches. However, the argument of latitude moves with an almost constant velocity to its reference and is, therefore, easy to predict. Therefore, the following absolute value:

$$|\delta\lambda_2^f - \delta\lambda_1^f - \lambda_{\text{coll}}^f|,$$

  where λ_{coll}^f is defined as:

$$\lambda_{\text{coll}}^f := \theta_{\text{coll}}^2 - \theta_{\text{coll}}^1 + \cos i(\Omega_2 - \Omega_1) - \lambda_{\text{ref},2}^f + \lambda_{\text{ref},1}^f,$$

  is approximately equal to the following:

$$\delta\lambda_2^f - \delta\lambda_1^f - \lambda_{\text{coll}}^f + \lambda_{\text{abs}}^f,$$

  where λ_{abs}^f is defined as follows, using the resulting value of $\delta\lambda_2^f - \delta\lambda_1^f - \lambda_{\text{coll}}^f$ from the previous MPC iteration:

$$\lambda_{\text{abs}}^f := \begin{cases} 0, & \text{if } \delta\lambda_2^f - \delta\lambda_1^f - \lambda_{\text{coll}}^f \geq 0, \\ -2(\delta\lambda_2^f - \delta\lambda_1^f - \lambda_{\text{coll}}^f), & \text{if } \delta\lambda_2^f - \delta\lambda_1^f - \lambda_{\text{coll}}^f < 0. \end{cases}$$

Secondly, the constraint makes both explicitly and implicitly (through θ_{coll}) use of Ω, which is not available as one of the state variables. However, one can use ξ_x and ξ_y as defined in Section 2-2-3:

$$\xi_x = \cos\theta_d \tan\frac{i_d}{2} - \cos(\theta_d + \Omega_d - \Omega_c) \tan\frac{i_c}{2},$$
$$\xi_y = \sin\theta_d \tan\frac{i_d}{2} - \sin(\theta_d + \Omega_d - \Omega_c) \tan\frac{i_c}{2}.$$

Assuming that the inclination does not change and using the trigonometric sum identities, this can be rewritten as:

$$\xi_x = (\cos\theta_d - \cos\theta_d \cos(\Omega_d - \Omega_c) - \sin\theta_d \sin(\Omega_d - \Omega_c)) \tan\frac{i_c}{2},$$
$$\xi_y = (\sin\theta_d - \sin\theta_d \cos(\Omega_d - \Omega_c) + \cos\theta_d \sin(\Omega_d - \Omega_c)) \tan\frac{i_c}{2},$$

which, after careful working out, yields the following relationship:

$$\xi_x^2 + \xi_y^2 = (2 - 2\cos(\Omega_d - \Omega_c)) \tan\frac{i_c}{2}.$$

Finding the corresponding Ω_d then is simple, although one has to account for the sign of $\Omega_d - \Omega_c$ as arccos only returns values in the range $[0, \pi]$. This is done by looking at the initial value of $\Omega_d^0 - \Omega_c^0$, as Ω_d is monotonically steered towards Ω_c as shown in Chapter 2. Therefore, Ω_d is found as:

$$\Omega_d = \begin{cases} \arccos\left(1 - \frac{\xi_x^2 + \xi_y^2}{2\tan\frac{i_c}{2}}\right) + \Omega_c, & \text{if } \Omega_d^0 - \Omega_c^0 \geq 0, \\ -\arccos\left(1 - \frac{\xi_x^2 + \xi_y^2}{2\tan\frac{i_c}{2}}\right) + \Omega_c, & \text{if } \Omega_d^0 - \Omega_c^0 < 0. \end{cases}$$

As this result is nonlinear, it cannot directly be applied in the constraint. Therefore, the state vector from the previous MPC iteration is used to compute Ω_d. These changes lead to the following nominal constraint:

$$r_{\text{sign}}(\lambda_2^{f,0}, \lambda_1^{f,0}) \cdot (\delta r_2 - \delta r_1) + \alpha_{\text{w}}(\delta\lambda_2^f - \delta\lambda_1^f - \lambda_{\text{coll}}^f + \lambda_{\text{abs}}^f) \geq r_{\min}. \tag{5-6}$$

Rewriting this constraint into the robust variant follows the same steps as the in-plane collision constraint in Section 5-2-1.

Constraint Selection

The constellation is set up such that the constraints are inactive at the start and end of the simulation. Satellite pairs that start with a positive value of the angular part of the collision avoidance constraint, i.e.:

$$\delta\lambda_2^f - \delta\lambda_1^f - \lambda_{\text{coll}}^f > 0,$$

and which also end with a negative value of this constraint, have a constraint added for this pair. The same statement holds for the opposite, where satellites start with a negative value for the equation above and where it is positive at the end. Mathematically, this can be expressed as:

$$\begin{aligned}
&(i,j) \in \mathcal{C}_{\text{start}}^{\text{pos}} \mid \delta\lambda_2^f(0) - \delta\lambda_1^f(0) - \lambda_{\text{coll}}^f > 0, \\
&(i,j) \in \mathcal{C}_{\text{start}}^{\text{neg}} \mid \delta\lambda_2^f(0) - \delta\lambda_1^f(0) - \lambda_{\text{coll}}^f < 0, \\
&(i,j) \in \mathcal{C}_{\text{end}}^{\text{pos}} \mid \delta\lambda_{\text{ref},2}^f - \delta\lambda_{\text{ref},1}^f - \lambda_{\text{coll}}^f > 0, \\
&(i,j) \in \mathcal{C}_{\text{end}}^{\text{neg}} \mid \delta\lambda_{\text{ref},2}^f - \delta\lambda_{\text{ref},1}^f - \lambda_{\text{coll}}^f < 0, \\
&(i,j) \in \mathcal{C}_{\text{total}} \mid \ ((i,j) \in \mathcal{C}_{\text{start}}^{\text{pos}} \wedge (i,j) \in \mathcal{C}_{\text{end}}^{\text{neg}}) \vee ((i,j) \in \mathcal{C}_{\text{start}}^{\text{neg}} \wedge (i,j) \in \mathcal{C}_{\text{end}}^{\text{pos}})
\end{aligned}$$

5-3 Large-Scale Simulation

To get a clear overview of the effect of the constraints, disturbances and different controllers, the simulation is run for four different setups:

- A simulation with a nominal controller without disturbances or collision avoidance constraints.
- A simulation with a nominal controller without any disturbances but with collision avoidance constraints.

- A simulation with a nominal controller with disturbances and collision avoidance constraints.
- A simulation with a robust controller with disturbances and collision avoidance constraints.

These scenarios become increasingly more difficult but more realistic at the same time as well. The results for the large-scale simulation are divided into three different sections. First, an overview of the reconfiguration procedure is provided in Section 5-3-1, after which the in-plane and out-of-plane constraints are analysed in Section 5-3-2 and Section 5-3-3, respectively.

5-3-1 Reconfiguration

Fig. 5-4 shows an overview of the movements of all 225 satellites in the simulation, similar to the planned path shown in Fig. 5-1. This figure corresponds to the data of the robust controller with disturbances and collision avoidance constraints, but all setups produce almost identical results from this view. Due to the disturbance, very small errors are visible when zooming in, but the general view is always visually identical to Fig. 5-4.

Firstly, it is important to note that all values for θ are plotted in the range $(-180, 180]$ degrees, whereas the values for Ω are plotted in the range $[0, 360)$ degrees. This causes the jumps in the plot and the lines that do not directly seem to have a starting or ending location specified.

Secondly, the satellites do not always move in a straight line to their final position because they first tend to decrease their value for $\delta\lambda^f$, and afterwards direct $\delta\xi_x$ and $\delta\xi_y$ to zero. Those satellites follow the contour lines for a constant λ^f as drawn in Fig. 5-2.

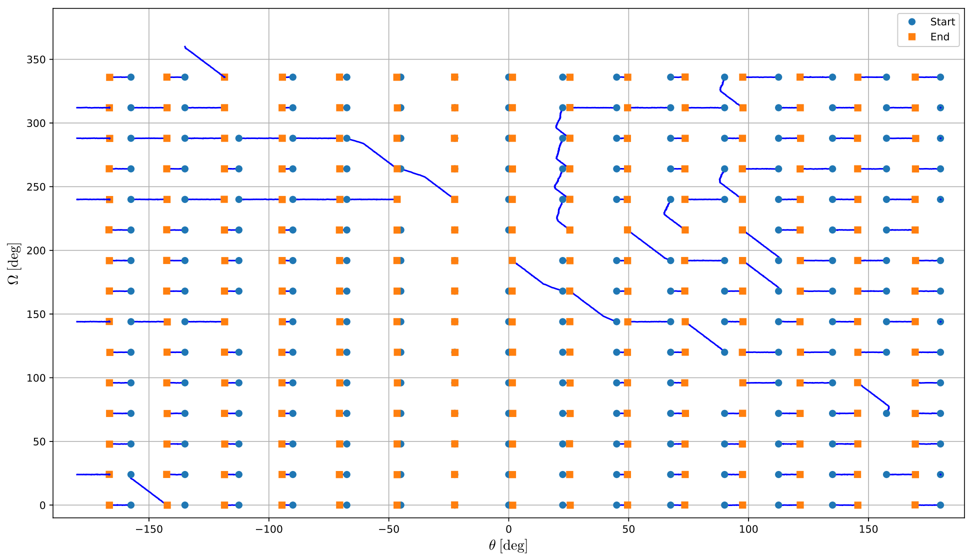


Figure 5-4: Reconfiguration overview of large-scale simulation.

5-3-2 In-Plane Collision Avoidance

The value of the left-hand side of the in-plane collision avoidance constraint of Eq. (5-2) is plotted in Fig. 5-5 for all tuples in $\mathcal{C}_{\text{total}}$ as defined in Section 5-2-1. The red dashed line denotes the minimum value of that constraint, which is set to five degrees. As the differences close to the constraint are difficult to see at this scale, Fig. 5-6 shows the area around the constraint. Both these plots show the result for the robust controller, but all controllers show a similar result. The main difference is that the simulations without the disturbances show smooth lines with less oscillations, but the values remain largely the same.

The large values of specifically one constraint may initially seem an error. As this constraint is only applied between neighbours, the difference between the values for λ^f does not seem to be able to be more than 100 degrees at all times. However, this is a special case where the plane with an Ω of 216 degrees at the start misses four satellites in a row, see Fig. 5-1. This results in two 'neighbours' that are relatively far apart.

Finally, it should be noted that the constraint is not active in this scenario, at least not without considering the uncertainty. It is possible that the robust controller was required to find a different solution because, due to uncertainty in the model, it could not guarantee the constraint at the start or end of the simulation. Because the initial value of the constraint is relatively low, it is not possible to increase the minimum value of the constraint in this scenario either.

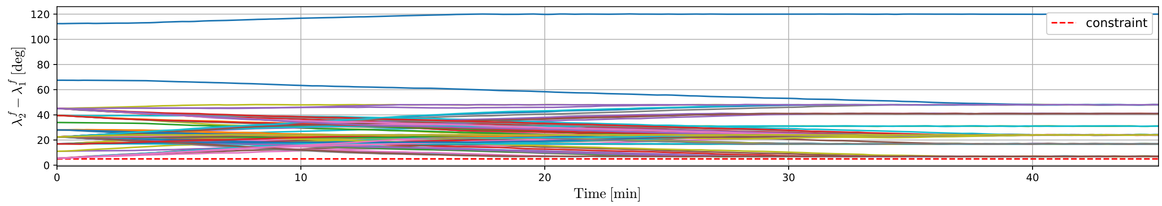


Figure 5-5: Full overview of in-plane collision avoidance constraint values.

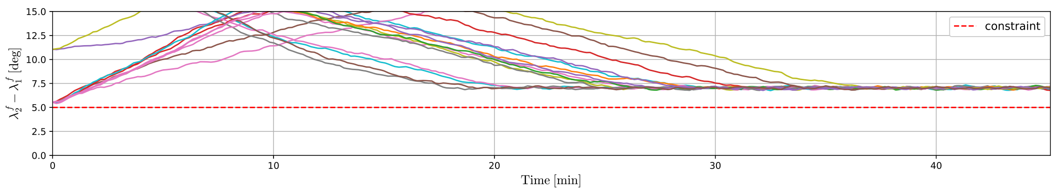


Figure 5-6: Overview of in-plane collision avoidance constraint values around the minimal value.

5-3-3 Out-Of-Plane Collision Avoidance

The left-hand side of the out-of-plane collision avoidance constraint from Eq. (5-4) is shown for all possible crossings in Fig. 5-7 for the simulation where the constraint is not enforced. As the critical constraint value details are hard to see, Fig. 5-8 provides a closer overview. It is clear that several satellites are violating the constraint, some even reaching a distance of zero meters and, thus, a collision.

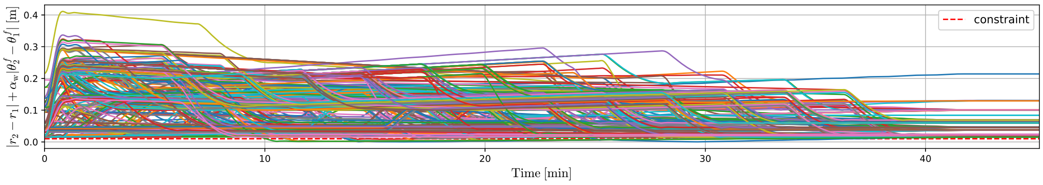


Figure 5-7: Out-of-plane collision avoidance constraint without enforcement.

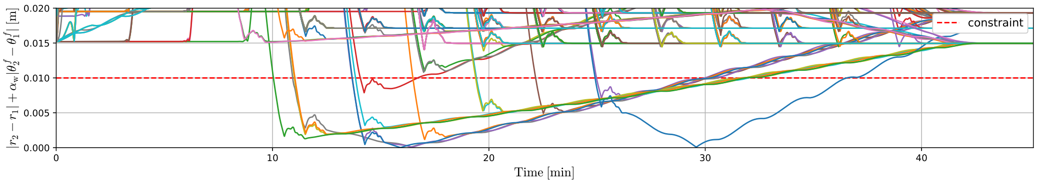


Figure 5-8: Out-of-plane collision avoidance constraint around limit without enforcement.

This can be compared to Fig. 5-9, where there are no disturbances in the simulation but the nominal controller now enforces the constraint. Despite reformulating the problem such that the absolute values are removed in the formulation for the controller, the controller can enforce the constraint very accurately. The values shown for all collision avoidance constraints are calculated using the exact equations (i.e., Eq. (5-4)) and not the approximated values used by the controllers (i.e., Eq. (5-6)).

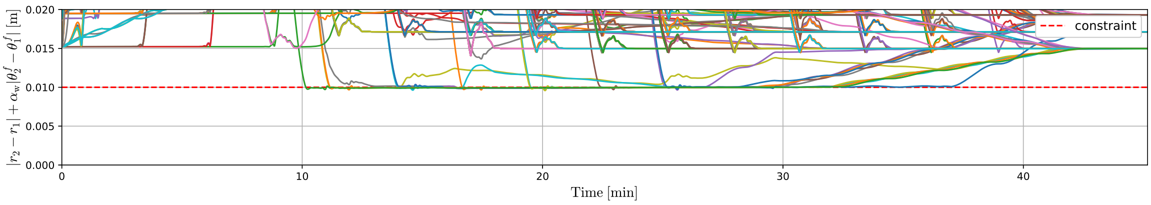


Figure 5-9: Out-of-plane collision constraint for nominal controller without disturbances.

The distance between the satellites is larger because the controller forces the radial states away from zero. This can be seen by comparing Fig. 5-10, where all radial states move directly to zero, with Fig. 5-11, where a radial offset is forced to guarantee that the satellites do not collide. At the end of the simulation, all satellites have reached their desired radius.

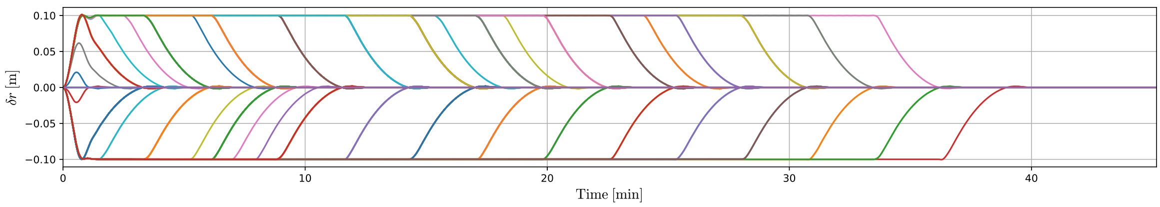


Figure 5-10: Radial state for nominal controller without collision constraint.

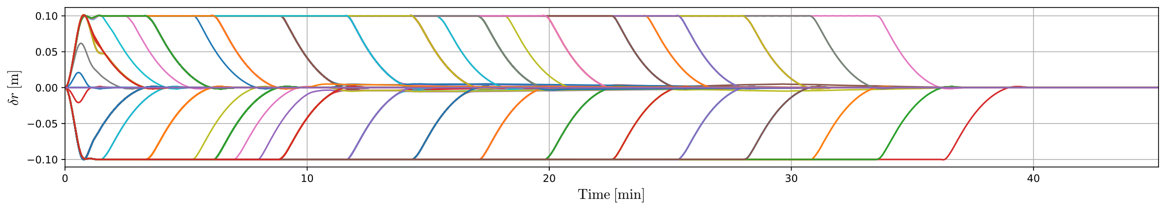


Figure 5-11: Radial state for nominal controller with collision constraint.

The nominal controller works well without any disturbances. However, once these are added, the controller violates the constraints again as shown in Fig. 5-12. Therefore, a robust controller is used to guarantee that the constraints are always met, and these results are shown in Fig. 5-13. The difference with the nominal controller is clear, as the robust controller stays clear from the critical constraint value and never violates the constraint.

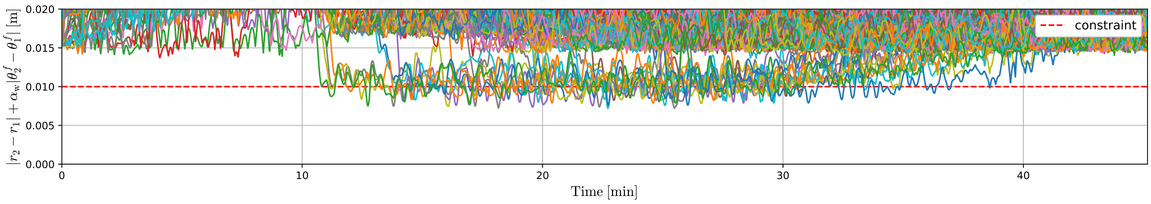


Figure 5-12: Out-of-plane collision avoidance constraint for nominal controller with disturbances.

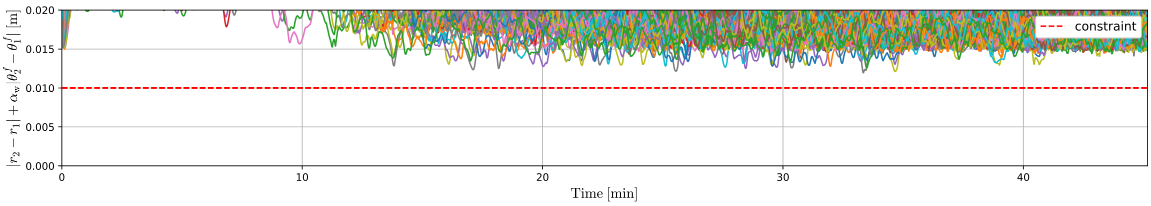


Figure 5-13: Out-of-plane collision avoidance constraint for robust controller.

5-4 Conclusions

The goal of this chapter was to show that the results from the previous chapters can be combined to control a large-scale constellation robustly while at the same time providing collision avoidance constraints to guarantee a safe distance between the satellites.

The simple assignment algorithm based on the Hungarian algorithm proved to work sufficiently, where through a simple cost function, the satellites were given a reference position that was efficient to reach and worked well together with the in-plane collision avoidance constraint.

The in-plane collision avoidance constraint is simple but can work well even for satellites that do not start within the same plane. The constraint can be used to space the satellites out during the reconfiguration if desired, but the starting and ending positions can limit this as these inherently limit the maximum value of this constraint.

The out-of-plane collision avoidance constraint is more difficult, but with several tricks can be rewritten into a simple linear constraint. The simulations show that the satellites keep a larger distance between themselves by increasing their radial difference and that even with disturbances, the robust controller can guarantee a safe distance.

With the large-scale simulation working, it is now possible to answer the research question and conclude the work in Chapter 6.

Chapter 6

Conclusion

This work is concluded by two different parts. First, the research questions stated in Chapter 1 are answered in Section 6-1. Afterwards, possible directions for future work are discussed in Section 6-2.

6-1 Research Questions

First, the sub-questions are answered in Section 6-1-1. The answers to the sub-questions lead to the answer of the main research question in Section 6-1-2.

6-1-1 Sub-Questions

> *What dynamical model best controls satellites for both in-plane and out-of-plane movements?*

Due to the scale of the problem, a LTI model is preferred, as well as a model that can work well for both in-plane and out-of-plane movements. Furthermore, having the radius as a state is beneficial for the cost function and the collision avoidance constraint, and the possibility to add different perturbations, such as J_2 perturbation, can add to the model's accuracy.

As no such model was readily available in the literature, a new model was developed that meets all these constraints. This new model has been shown to work well in various scenarios and has taken inspiration from the cylindrical HCW model and the quasi-nonsingular ROE model. The state variables are substantially different, however.

> *What control algorithm can be used best to control a satellite robustly?*

There are many possible robust control algorithms, but a method that stood out was the lumped SLS control method as outperformed tube MPC in, for example, both speed and conservatism. Where SLS optimises closed-loop transfer functions, the lumped refers to the fact that both model uncertainties and disturbances are lumped together as a single disturbance.

The original lumped SLS formulation could sometimes be too conservative, which is the reason a modification is proposed to reduce this. This modification has been proposed before, but not without any assumptions on the structure of the model uncertainty and without the number of constraints scaling exponentially in those cases. The modification has been shown to guarantee meeting all constraints successfully.

How can a different formulation of the optimisation problem scale up the control algorithm?

The robust SLS formulation used nonlinear constraints containing the one-norm and the infinity-norm. For larger problems, this slows down the speed of the problem significantly. Using several dummy variables, it is possible to rewrite these constraints linearly. Normally, an extra cost must be added to ensure these slack variables find the correct values. However, as they represent the uncertainty in the model, an inherent force decreases these dummy variables until they have the correct value.

Furthermore, a significant part of the closed-loop transfer function is known to be zero beforehand. Another significant speed increase is obtained by removing these values from the optimisation problem and by reformulating the problem to a standard quadratic problem. The fastest solver for the robust problem is `Gurobi`.

How can collision avoidance constraints be (robustly) formulated with this dynamical model and control algorithm?

Two types of collision avoidance constraints are added. The first deals with in-plane collisions, which is a relatively easy constraint. Although no direct state represents the argument of latitude, the state λ^f represents a combination of the argument of latitude and the RAAN, which allows it to work for satellites that do not start in the same plane.

The second constraint deals with the satellites within different planes. These satellites can still collide because they are both in orbits on the same sphere and therefore, these orbits have to intersect each other at two points. The constraint combines radial and angular distances to guarantee a safe crossing.

6-1-2 Main Research Question

The main research question was as follows:

How can a large constellation of several hundreds of satellites and multiple planes be robustly controlled while maintaining a safe distance between themselves?

The answer combines the use of the newly developed model with lumped SLS as the control algorithm, using the modification from this work to reduce conservatism. To allow for the large scale, the optimisation problem rewrites the nonlinear constraints as linear ones and uses a sparse vector variant. Both in-plane and out-of-plane collision avoidance constraints can be used to guarantee a safe distance between the satellites.

6-2 Future Work

Naturally, there a many possibilities to extend upon this work in future work:

- The new model can be extended to allow for more types of disturbances to be included in the model, such as drag forces.
- Especially the effect of the eccentricity of the orbit could be analysed further. It might be possible to create a variant of the Blend model that is also valid for more eccentric orbits, similar to the Lawden-Tschauner-Hempel (LTH) models.

- The modification for the lumped SLS method can be extended to polytopic disturbances.
- The sources of uncertainty from the modelling errors, such as the assumption that the orbit is circular, could be used to find structure in the modelling errors and thus reduce the conservatism of the robust controller.
- The use of a GPU to solve large problems faster could be investigated in more detail. The use of `cuOSQP` seemed promising due to the high speed for most problems, but it failed to outperform the other solvers for the final SLS problem. A different factorisation could help.
- Similarly, the use of `OSQP` could be investigated further. The solver produces promising results for nominal SLS problems but fails to find accurate results for the robust problems.
- Finally, one could work on a receding horizon implementation by formally providing recursive feasibility and stability with the new robust MPC.

Appendix A

Model Derivations

This appendix contains several model derivations, such as those of the cylindrical HCW model in Appendix A-1, the quasi-nonsingular ROE model in Appendix A-2 and the Blend model in Appendix A-3.

A-1 Hill-Clohessy-Wiltshire Model

Starting from Eq. (2-8) with only the control inputs as disturbances, the problem can be transformed into the cylindrical coordinates $[r, \varphi, z]$:

$$\begin{aligned}\ddot{r} - r\dot{\varphi}^2 + n^2 r &= u_r,\\ r\ddot{\varphi} + 2\dot{r}\dot{\varphi} &= u_t,\\ \ddot{z} + n^2 z &= u_n,\end{aligned} \tag{A-1}$$

where

$$n = \sqrt{\frac{\mu}{(r^2 + z^2)^{\frac{3}{2}}}}.$$

The control inputs $[u_r, u_t, u_n]$ are accelerations in the radial, tangential and normal directions, respectively. When the reference orbit is circular with an orbital radius of r_c and given that $z_c = \dot{z}_c = \ddot{z}_c = 0$ and $\dot{\varphi}_c = n_c$, a relative model can be obtained:

$$\begin{aligned}\Delta\ddot{r} &= \ddot{r}_d - \ddot{r}_c\\ &= r_d\dot{\varphi}_d^2 - n_d^2 r_d + u_{r,d} - r_c\dot{\varphi}_c^2 + n_c^2 r_c\\ &= (r_c + \Delta r)(n_c + \Delta\dot{\varphi})^2 - \frac{\mu(r_c + \Delta r)}{((r_c + \Delta r)^2 + (\Delta z)^2)^{\frac{3}{2}}} + u_r,\\ \\ \Delta\ddot{\varphi} &= \ddot{\varphi}_d - \ddot{\varphi}_c\\ &= -2\frac{\dot{r}_d}{r_d}\dot{\varphi}_d + \frac{u_{t,d}}{r_c} + 2\frac{\dot{r}_c}{r_c}\dot{\varphi}_c\\ &= -2\frac{\Delta\dot{r}(n_c + \Delta\dot{\varphi})}{r_c + \Delta r} + \frac{u_t}{r_c},\\ \\ \Delta\ddot{z} &= \ddot{z}_d - \ddot{z}_c\\ &= -n_d^2 z_d + u_n + n_c^2 z_c\\ &= -\frac{\mu\Delta z}{((r_c + \Delta r)^2 + (\Delta z)^2)^{\frac{3}{2}}} + u_n.\end{aligned} \tag{A-2}$$

Assuming $\Delta r, \Delta z \ll r_c$ and $\Delta\dot{\varphi} \ll \dot{\varphi}_c$, Eq. (A-2) yields after linearisation through a first-order Taylor expansion:

$$\begin{aligned}
\Delta\ddot{r} - 3n_c^2\Delta r - 2r_c n_c \Delta\dot{\varphi} &= u_r, \\
\Delta\ddot{\varphi} + \frac{2n_c}{r_c}\Delta\dot{r} &= \frac{u_t}{r_c}, \\
\Delta\ddot{z} + n_c^2\Delta z &= u_n.
\end{aligned}$$

A-2 Quasi-Nonsingular Relative Orbital Elements Model

There are three different contributions to the quasi-nonsingular ROE model: Keplerian forces (Section A-2-1), J_2 perturbations (Section A-2-2) and control inputs (Section A-2-3).

A-2-1 Keplerian Dynamics

In the case of perfect Keplerian dynamics, the derivative of all orbital elements is zero, except for M, where $\dot{M} = n = \sqrt{\frac{\mu}{a^3}}$. Linearising this through a first-order Taylor expansion around $\dot{M}_c$ yields the following result for $\dot{M}_d$:

$$\begin{aligned}
\dot{M}_d &= \sqrt{\frac{\mu}{a_d^3}} \\
&\approx \sqrt{\frac{\mu}{a_c^3}} - \frac{3}{2}\sqrt{\frac{\mu}{a_c^3}}\frac{a_d - a_c}{a_c},
\end{aligned}$$

or in a relative form:

$$\begin{aligned}
\delta\dot{\lambda} &= \dot{M}_d - \dot{M}_c \\
&\approx \sqrt{\frac{\mu}{a_c^3}} - \frac{3}{2}\sqrt{\frac{\mu}{a_c^3}}\frac{a_d - a_c}{a_c} - \sqrt{\frac{\mu}{a_c^3}} \\
&= -\frac{3}{2}n_c\delta a.
\end{aligned}$$

A-2-2 J_2 Perturbations

In [40] the authors use the same state vector as in Eq. (2-16), and they are able to directly derive the effect of J_2 perturbations on this model:

$$\begin{aligned}
\delta\dot{\lambda} &= -\frac{21}{4}\beta((3\cos^2 i - 1)\eta + 5\cos^2 i - 1)\,\delta a - \frac{3}{2}\beta\sin 2i(3\eta + 5)\,\delta i_x, \\
\delta\dot{e}_x &= -\frac{3}{2}\beta(5\cos^2 i - 1)\,\delta e_y, \\
\delta\dot{e}_y &= \frac{3}{2}\beta(5\cos^2 i - 1)\,\delta e_x, \\
\delta\dot{i}_y &= \frac{21}{4}\beta\sin 2i\,\delta a + 3\beta\sin^2 i\,\delta i_x,
\end{aligned}$$

where

$$\beta = \frac{1}{2}\frac{J_2 R_e^2 n}{a^2\eta^4}.$$

A-2-3 Control Inputs

The effect of the control inputs as provided in Eq. (2-14) is on the orbital elements themselves, where the state vector in Eq. (2-16) contains nonlinear combinations of these elements. The effect of the control inputs on some of these states is provided in [41]:

$$\begin{aligned}
\frac{d\,a}{dt} &= 2a\frac{u_t}{na}, \\
\frac{d\,M+\omega}{dt} &= -2\frac{u_r}{na} - \frac{\sin(M+\omega)}{\tan i}\frac{u_n}{na}, \\
\frac{d\,e_x}{dt} &= -2\cos(M+\omega)\frac{u_t}{na} + \sin(M+\omega)\frac{u_r}{na}, \\
\frac{d\,e_y}{dt} &= -2\sin(M+\omega)\frac{u_t}{na} - \cos(M+\omega)\frac{u_r}{na}, \\
\frac{d\,i}{dt} &= \cos(M+\omega)\frac{u_n}{na}, \\
\frac{d\,\Omega}{dt} &= \frac{\sin(M+\omega)}{\sin i}\frac{u_n}{na},
\end{aligned}$$

such that effect of the control inputs for all states in Eq. (2-16) can be found using these equations by simple additions and multiplications:

$$\begin{aligned}
\delta\dot{a} &= 2\frac{u_t}{na}, \\
\delta\dot{\lambda} &= -2\frac{u_r}{na}, \\
\delta\dot{e}_x &= -2\cos(M+\omega)\frac{u_t}{na} + \sin(M+\omega)\frac{u_r}{na}, \\
\delta\dot{e}_y &= -2\sin(M+\omega)\frac{u_t}{na} - \cos(M+\omega)\frac{u_r}{na}, \\
\delta\dot{i}_x &= \cos(M+\omega)\frac{u_n}{na}, \\
\delta\dot{i}_y &= \sin(M+\omega)\frac{u_n}{na}.
\end{aligned}$$

A-3 Blend Model

The derivation of the Blend model is divided into five parts. First, the derivative for the angular state $\delta\lambda^f$ is given in Section A-3-1, followed by the derivatives for the radial state δr (Section A-3-2), for the eccentric states δe_x^f and δe_y^f (Section A-3-3) and for the out-of-plane states $\delta\xi_x$ and $\delta\xi_y$ (Section A-3-4). The section ends with the extension of the model to include J_2 perturbations in Section A-3-5.

A-3-1 Angular State

The angular state of the model has to keep in mind that, as briefly discussed at the end of Section 2-1-2, not all orbital parameters are well-defined for circular orbits. More specifically, all three anomalies and the argument of periapsis are not well-defined due to the lack of a line of apsides.

However, a simple workaround is to add the argument of periapsis to one of these anomalies, as this produces a valid result. As the goal is to find a model that can be used for collision avoidance constraints, ideally, the argument of latitude as defined in Eq. (2-7) is used for the angular state as this represents the actual angle, as opposed to a fictitious angle with the mean argument of latitude that is used for the quasi-singular ROE. However, as will become apparent later, simply using the argument of latitude as a state variable results in a LTV system. Therefore, the angular state variable λ^f for a single satellite is defined as:

$$\lambda^f := f + \omega + \Omega \cos i = \theta + \Omega \cos i.$$

Finding the derivative of the second term is often simple, as it is assumed that the inclination of the chief satellite is close to the inclination of the deputy satellite and remains (almost) constant. This means that the derivative, using Eq. (2-14), is simply equal to:

$$\frac{d\,\Omega \cos i}{dt} \approx \dot{\Omega} \cos i = \frac{\sqrt{1-e^2}\sin\theta}{an(1+e\cos f)\tan i} u_n. \tag{A-3}$$

Finding the derivative of θ can be tricky if done incorrectly. Two approaches are presented here: a naive and a preferred approach. The relative model follows this as a result.

Naive approach

As it is assumed that the eccentricity is small, a glance at Eq. (2-5) seems to suggest that the true anomaly is simply equal to the mean anomaly, and therefore, the derivatives are equal as well:

$$f + \omega \approx M + \omega \implies \dot{f} + \dot{\omega} \approx \dot{M} + \dot{\omega}.$$

This would result in an angular state that is close to the ROE state, and its derivative can be found using Eq. (2-14):

$$\begin{aligned}
\dot{M} + \dot{\omega} = n &+ \frac{(1-e^2)\left[(1+e\cos f)\cos f - 2e\right]}{ane(1+e\cos f)} u_r - \frac{(1-e^2)(2+e\cos f)\sin f}{ane(1+e\cos f)} u_t \\
&- \frac{\sqrt{1-e^2}\cos f}{ane} u_r + \frac{\sqrt{1-e^2}(2+e\cos f)\sin f}{ane(1+e\cos f)} u_t - \frac{\sqrt{1-e^2}\sin\theta}{an(1+e\cos f)\tan i} u_n \\
&= n + \frac{\left((1-e^2) - \sqrt{1-e^2}\right)(1+e\cos f)\cos f - 2e(1-e^2)}{ane(1+e\cos f)} u_r \\
&+ \frac{\left(\sqrt{1-e^2} - (1-e^2)\right)(2+e\cos f)\sin f}{ane(1+e\cos f)} u_t - \frac{\sqrt{1-e^2}\sin\theta}{an(1+e\cos f)\tan i} u_n.
\end{aligned}$$

Note how the terms before u_r and u_t cause problems when the eccentricity tends to zero, resulting in dividing by zero. Using l'Hôpital's rule, however, it is possible to find the limit for the eccentricity approaching zero:

$$
\begin{aligned}
&\lim_{e\to 0}\frac{\left((1-e^2)-\sqrt{1-e^2}\right)(1+e\cos f)\cos f-2e(1-e^2)}{ane(1+e\cos f)}\\
=&\lim_{e\to 0}\frac{\frac{\partial}{\partial e}\left[\left((1-e^2)-\sqrt{1-e^2}\right)(1+e\cos f)\cos f-2e(1-e^2)\right]}{\frac{\partial}{\partial e}\left[ane(1+e\cos f)\right]}\\
=&\lim_{e\to 0}\frac{\left(-2e+\frac{e}{\sqrt{1-e^2}}\right)(1+e\cos f)\cos f+\left((1-e^2)-\sqrt{1-e^2}\right)\cos^2 f-2(1-e^2)+4e^2}{an(1+e\cos f)+ane\cos f}\\
=&\frac{-2}{an},
\end{aligned}
$$

$$
\begin{aligned}
&\lim_{e\to 0}\frac{\left(\sqrt{1-e^2}-(1-e^2)\right)(2+e\cos f)\sin f}{ane(1+e\cos f)}\\
=&\lim_{e\to 0}\frac{\frac{\partial}{\partial e}\left[\left(\sqrt{1-e^2}-(1-e^2)\right)(2+e\cos f)\sin f\right]}{\frac{\partial}{\partial e}\left[ane(1+e\cos f)\right]}\\
=&\lim_{e\to 0}\frac{\left(-\frac{e}{\sqrt{1-e^2}}+2e\right)(2+e\cos f)\cos f+\left(\sqrt{1-e^2}-(1-e^2)\right)\sin f\cos f}{an(1+e\cos f)+ane\cos f}\\
=&0.
\end{aligned}
$$

Thus, with the naive approach, the derivative of the angular state for orbits with a small eccentricity can be found by combining this result with Eq. (A-3), which yields:

$$
\begin{aligned}
\dot{\lambda}^f &\approx \dot{M}+\dot{\omega}+\dot{\Omega}\cos i\\
&= n-\frac{2}{an}u_r-\frac{\sin\theta}{\tan i}u_n+\frac{\sin\theta}{\tan i}u_n\\
&= n-\frac{2}{an}u_r.
\end{aligned}
\tag{A-4}
$$

Note that this approach essentially equals the state variable used for the quasi-nonsingular ROE. If one would create a relative model out of Eq. (A-4), the same terms as in Appendix A-2 would appear.

Preferred Approach

When working with these small eccentricities, it is best to make the assumptions as late as possible in the derivations. For example, when the argument of latitude is computed with the actual derivative of the true anomaly:

$$
\begin{aligned}
\dot{f} &= \frac{h}{r^2}+\frac{p}{he}\cos f u_r-\frac{p+r}{he}\sin f\; u_t\\
&= \frac{n(1+e\cos f)^2}{(1-e^2)^{\frac{3}{2}}}+\frac{\sqrt{1-e^2}\cos f}{ane}u_r-\frac{\sqrt{1-e^2}(2+e\cos f)\sin f}{ane(1+e\cos f)}u_t,
\end{aligned}
\tag{A-5}
$$

the effect of the radial input, which has a large impact in the naive approach, completely vanishes:

$$
\begin{aligned}
\dot{\theta} &= \dot{f} + \dot{\omega} \\
&= \frac{n(1+e\cos f)^2}{(1-e^2)^{\frac{3}{2}}} + \frac{\sqrt{1-e^2}\cos f}{ane}u_r - \frac{\sqrt{1-e^2}(2+e\cos f)\sin f}{ane(1+e\cos f)}u_t \\
&- \frac{\sqrt{1-e^2}\cos f}{ane}u_r + \frac{\sqrt{1-e^2}(2+e\cos f)\sin f}{ane(1+e\cos f)}u_t - \frac{\sqrt{1-e^2}\sin\theta}{an(1+e\cos f)\tan i}u_n \\
&= \frac{n(1+e\cos f)^2}{(1-e^2)^{\frac{3}{2}}} - \frac{\sqrt{1-e^2}\sin\theta}{an(1+e\cos f)\tan i}u_n \\
&= \underbrace{\sqrt{\frac{\mu(1+e\cos f)}{r^3}}}_{\dot{\theta}_f} + \underbrace{\frac{-\sqrt{1-e^2}\sin\theta}{an(1+e\cos f)\tan i}u_n}_{\dot{\theta}_\omega} .
\end{aligned}
\tag{A-6}
$$

The derivative of λ^f is then equal to:

$$
\begin{aligned}
\dot{\lambda}^f &= \dot{f} + \dot{\omega} + \dot{\Omega}\cos i \\
&= \sqrt{\frac{\mu(1+e\cos f)}{r^3}} - \frac{\sqrt{1-e^2}\sin\theta}{an(1+e\cos f)\tan i}u_n + \frac{\sqrt{1-e^2}\sin\theta}{an(1+e\cos f)\tan i}u_n \\
&= \sqrt{\frac{\mu(1+e\cos f)}{r^3}},
\end{aligned}
\tag{A-7}
$$

for which no assumptions on the eccentricity have been made yet. Also, note the importance of $\dot{\Omega}\cos i$, as it cancels the LTV term from $\dot{\theta}$.

Final Relative Result

When computing the relative states with respect to the reference satellite, a critical trick is used to increase the accuracy of the solution. Where normally the nonlinear equations are linearised through a first-order Taylor expansion around the chief satellite (i.e., around r_c, e_c and f_c), the nonlinear equations are here linearised around r_c, e_c and f_d[1]. As the reference satellite is in a circular orbit, its (second) derivatives are independent of the true anomaly and thus, no difference is obtained this way.

This new linearisation point does give an important advantage. For example, when com-

[1] This can be interpreted around linearising around a virtual satellite on the reference orbit, but with the same true anomaly as the actual satellite.

puting the relative angular state from A-7:

$$
\begin{aligned}
\delta\dot{\lambda}^f &= \dot{\lambda}_d^f - \dot{\lambda}_c^f \\
&= \sqrt{\frac{\mu(1+e_d\cos f_d)}{r_d^3}} - \sqrt{\frac{\mu(1+e_c\cos f_c)}{r_c^3}} \\
&\approx -\frac{3}{2}\sqrt{\frac{\mu(1+e_c\cos f_d)}{r_c^3}}\frac{r_d-r_c}{r_c} + \frac{1}{2}\sqrt{\frac{\mu(1+e_c\cos f_d)}{r_c^3}}\frac{\cos f_d}{1+e_c\cos f_d}(e_d-e_c) \\
&= -\frac{3}{2}\frac{n_c}{r_c}(r_d-r_c) + \frac{1}{2}n_c e_d\cos f_d \\
&= -\frac{3}{2}\frac{n_c}{r_c}\delta r + \frac{1}{2}n_c\delta e_x^f,
\end{aligned} \tag{A-8}
$$

where the last line holds because the reference orbit is assumed to be perfectly circular and where e_x^f is defined as:

$$e_x^f := e\cos f.$$

As $e_c\cos f_c$ is always zero, δe_x^f is equal to$e_d\cos f_d$.

A-3-2 Radial State

The derivative of the radius, as defined in Eq. (2-6), can easily be obtained:

$$\dot{r} = \frac{(1-e^2)}{1+e\cos f}\dot{a} - \left(\frac{2ae}{1+e\cos f} + \frac{a(1-e^2)\cos f}{(1+e\cos f)^2}\right)\dot{e} + \frac{ae(1-e^2)\sin f}{(1+e\cos f)^2}\dot{f}, \tag{A-9}$$

which can be combined with the results from Eqs. (A-5) and (2-14) to obtain the result in terms of the control inputs:

$$
\begin{aligned}
\dot{r} &= \frac{(1-e^2)}{1+e\cos f}\left(\frac{2e\sin f}{n\sqrt{1-e^2}}u_r + \frac{2(1+e\cos f)}{n\sqrt{1-e^2}}u_t\right) \\
&- \left(\frac{2ae}{1+e\cos f} + \frac{a(1-e^2)\cos f}{(1+e\cos f)^2}\right)\left(\frac{\sqrt{1-e^2}\sin f}{an}u_r + \frac{\sqrt{1-e^2}\left[(2+e\cos f)\cos f + e\right]}{an(1+e\cos f)}u_t\right) \\
&+ \frac{ae(1-e^2)\sin f}{(1+e\cos f)^2}\left(\frac{n(1+e\cos f)^2}{(1-e^2)^{\frac{3}{2}}} + \frac{\sqrt{1-e^2}\cos f}{ane}u_r - \frac{\sqrt{1-e^2}(2+e\cos f)\sin f}{ane(1+e\cos f)}u_t\right) \\
&= \frac{2e\sqrt{1-e^2}\sin f}{n(1+e\cos f)}u_r + 2\frac{\sqrt{1-e^2}}{n}u_t \\
&- \left(\frac{2e\sqrt{1-e^2}\sin f}{n(1+e\cos f)} + \frac{(1-e^2)^{\frac{3}{2}}\cos f\sin f}{n(1+e\cos f)^2}\right)u_r \\
&- \left(\frac{2e\sqrt{1-e^2}\left[(2+e\cos f)\cos f + e\right]}{n(1+e\cos f)^2} + \frac{(1-e^2)^{\frac{3}{2}}\left[(2+e\cos f)\cos^2 f + e\cos f\right]}{n(1+e\cos f)^3}\right)u_t \\
&+ \frac{ane\sin f}{\sqrt{1-e^2}} + \frac{(1-e^2)^{\frac{3}{2}}\sin f\cos f}{n(1+e\cos f)^2}u_r - \frac{(1-e^2)^{\frac{3}{2}}(2+e\cos f)\sin^2 f}{n(1+e\cos f)^3}u_t.
\end{aligned}
$$

While all terms for u_r cancel each other out, this is not directly obvious for u_t. However, currently left with:

$$\dot{r} = \frac{ane\sin f}{\sqrt{1-e^2}} + 2\left(\frac{\sqrt{1-e^2}}{n} - \frac{(1-e^2)^{\frac{3}{2}} + e\sqrt{1-e^2}\left[(2+e\cos f)\cos f + e\right]}{n(1+e\cos f)^2}\right)u_t,$$

it is possible to rewrite the numerator of the second term slightly:

$$\begin{aligned} e\left[(2+e\cos f)\cos f + e\right] &= (2+e\cos f)e\cos f + e^2 \\ &= (2+e\cos f)e\cos f + 1 - (1-e^2) \\ &= 1 + 2e\cos f + e^2\cos^2 f - (1-e^2) \\ &= (1+e\cos f)^2 - (1-e^2), \end{aligned}$$

such that the following result is obtained:

$$\begin{aligned} \dot{r} &= \frac{ane\sin f}{\sqrt{1-e^2}} + 2\left(\frac{\sqrt{1-e^2}}{n} - \frac{(1-e^2)^{\frac{3}{2}} + \sqrt{1-e^2}\left[(1+e\cos f)^2 - (1-e^2)\right]}{n(1+e\cos f)^2}\right)u_t \\ &= \frac{ane\sin f}{\sqrt{1-e^2}} \\ &= \sqrt{\frac{\mu}{a(1-e^2)}}e\sin f. \end{aligned} \tag{A-10}$$

This result is surprisingly simple after all the terms have been cancelled out. A few conclusions can already be drawn from this result. Firstly, unlike for the orbital elements with which the derivation started, the first-order derivative of the radius is not affected by the control inputs. This means that a second state is required to control the radius. Secondly, as expected, $\dot{r}$ is zero for perfectly circular orbits. The relative derivative is therefore easy to find, as $\dot{r}_c$ is zero:

$$\begin{aligned} \delta\dot{r} &= \dot{r}_d - \dot{r}_c \\ &= \sqrt{\frac{\mu}{a_d(1-e_d^2)}}e_d\sin f_d \\ &= \sqrt{\frac{\mu(1+e_d\cos f_d)}{r_d}}e_d\sin f_d \\ &\approx \sqrt{\frac{\mu(1+e_c\cos f_c)}{r_c}}e_c\sin f_d - \frac{1}{2}\sqrt{\frac{\mu(1+e_c\cos f_d)}{r_c}}e_c\sin f_d\frac{r_d-r_c}{r_c} \\ &+ \left(\frac{1}{2}\sqrt{\frac{\mu(1+e_c\cos f_d)}{r_c}}\frac{e_c\sin f_d}{1+e_c\cos f_d} + \sqrt{\frac{\mu(1+e_c\cos f_d)}{r_c}}\sin f_d\right)(e_d-e_c) \\ &= \sqrt{\frac{\mu(1+e_c\cos f_d)}{r_c}}e_d\sin f_d \\ &= n_c r_c \delta e_y^f. \end{aligned} \tag{A-11}$$

Here, the last line only holds because it is assumed that the reference orbit is perfectly circular and where e_y^f is defined as:

$$e_y^f := e\sin f.$$

As $e_c\sin f_c$ is always zero, δe_y^f equals $e_d\sin f_d$.

A-3-3 Eccentric States

The differential equations for the eccentric states are first derived, but several assumptions are made in the process. Next, the results are compared with the second derivatives to check if the errors in assumptions made in this and previous subsections do not compound. Finally, the results of the in-plane equations of motions are compared with the cylindrical HCW model.

Differentiation of Eccentric States

The two eccentric states, δe_x^f and δe_y^f, are added as states because they appear in Eqs. (A-8) and (A-11) for $\delta\dot{\lambda}^f$ and $\delta\dot{r}$, respectively. This also requires finding their derivatives, where the derivative of e_x^f is as follows:

$$\begin{aligned}
\dot{e}_x^f &= \dot{e}\cos f - e\dot{f}\sin f \\
&= \frac{\sqrt{1-e^2}\sin f\cos f}{an}u_r + \frac{\sqrt{1-e^2}\left[(2+e\cos f)\cos^2 f + e\cos f\right]}{an(1+e\cos f)}u_t \\
&- \frac{ne(1+e\cos f)^2\sin f}{(1-e^2)^{\frac{3}{2}}} - \frac{\sqrt{1-e^2}\sin f\cos f}{an}u_r + \frac{\sqrt{1-e^2}(2+e\cos f)\sin^2 f}{a(1+e\cos f)}u_t \\
&= 2\frac{\sqrt{1-e^2}}{an}u_t - \frac{n(1+e\cos f)^2}{(1-e^2)^{\frac{3}{2}}}e\sin f,
\end{aligned} \tag{A-12}$$

which, under the assumption that e is small, can be simplified to:

$$\dot{e}_x^f \approx \frac{2}{an}u_t - ne_y^f.$$

The derivative of e_y^f follows a similar procedure:

$$\begin{aligned}
\dot{e}_y^f &= \dot{e}\sin f + e\dot{f}\cos f \\
&= \frac{\sqrt{1-e^2}\sin^2 f}{an}u_r + \frac{\sqrt{1-e^2}\left[(2+e\cos f)\sin f\cos f + e\sin f\right]}{an(1+e\cos f)}u_t \\
&+ \frac{ne(1+e\cos f)^2\cos f}{(1-e^2)^{\frac{3}{2}}} + \frac{\sqrt{1-e^2}\cos^2 f}{an}u_r - \frac{\sqrt{1-e^2}(2+e\cos f)\sin f\cos f}{an(1+e\cos f)}u_t \\
&= \frac{\sqrt{1-e^2}}{an}u_r + \frac{\sqrt{1-e^2}e\sin f}{an(1+e\cos f)}u_t + \frac{na^2\sqrt{1-e^2}}{r^2}e\cos f,
\end{aligned} \tag{A-13}$$

although here an extra term is obtained. However, if again assuming that the eccentricity is small, this can be simplified to:

$$\dot{e}_y^f \approx \frac{1}{an}u_r + ne_x^f.$$

The relative dynamics of these states are relatively straightforward, as e_c is zero and thus $\delta\dot{e}_x^f = \dot{e}_{x,d}^f$ and $\delta\dot{e}_y^f = \dot{e}_{y,d}^f$. However, as a further simplification, in the following equations, it is assumed that the semi-major axes are roughly equal, and thus both a_d and n_d are approximately equal to a_c and n_c:

$$\begin{aligned}
\delta\dot{e}_x^f &= \dot{e}_{x,d}^f = \frac{2}{a_c n_c}u_t - n_c\delta e_y^f \\
\delta\dot{e}_y^f &= \dot{e}_{y,d}^f = \frac{1}{a_c n_c}u_r + n_c\delta e_x^f.
\end{aligned} \tag{A-14}$$

Comparison With Second Derivatives

The eccentric states are used to compute the derivatives for the radial and angular states, but both during the derivations of those equations (i.e., Eqs. (A-8) and (A-11)) and during the derivations of the differential equations for the eccentric states (i.e., Eq. (A-14)) assumptions were made. To check if the errors due to these assumptions do not compound, the second derivative of the radial and angular states are compared with the results from Eq. (A-14).

First, the second derivative of r is analysed. Therefore, the partial derivatives of Eq. (A-10) with respect to a, e and f are multiplied with their derivatives:

$$\ddot{r} = -\frac{1}{2}\sqrt{\frac{\mu}{a^3(1-e^2)}}e\sin f\dot{a} + \left(\frac{e^2}{1-e^2}+1\right)\sqrt{\frac{\mu}{a(1-e^2)}}\sin f\dot{e} + \sqrt{\frac{\mu}{a(1-e^2)}}e\cos f\dot{f},$$

where the second term can be simplified, as $\frac{e^2}{1-e^2}+1$ is equal to $\frac{1}{1-e^2}$, which can be moved into the square root. This result can be combined with Eq. (2-14) to find the effect of the control inputs:

$$\begin{aligned}
\ddot{r} &= -\frac{1}{2}\sqrt{\frac{\mu}{a^3(1-e^2)}}e\sin f\dot{a} + \sqrt{\frac{\mu}{a(1-e^2)^3}}\sin f\dot{e} + \sqrt{\frac{\mu}{a(1-e^2)}}e\cos f\dot{f} \\
&= -\frac{1}{2}\frac{n}{\sqrt{1-e^2}}e\sin f\dot{a} + \frac{na}{(1-e^2)^{\frac{3}{2}}}\sin f\dot{e} + \frac{na}{\sqrt{1-e^2}}e\cos f\dot{f} \\
\\
&= -\frac{1}{2}\frac{n}{\sqrt{1-e^2}}e\sin f\left(\frac{2e\sin f}{n\sqrt{1-e^2}}u_r + \frac{2(1+e\cos f)}{n\sqrt{1-e^2}}u_t\right) \\
&+ \frac{na}{(1-e^2)^{\frac{3}{2}}}\sin f\left(\frac{\sqrt{1-e^2}\sin f}{an}u_r + \frac{\sqrt{1-e^2}\left[(2+e\cos f)\cos f + e\right]}{an(1+e\cos f)}u_t\right) \\
&+ \frac{na}{\sqrt{1-e^2}}e\cos f\left(\frac{n(1+e\cos f)^2}{(1-e^2)^{\frac{3}{2}}} + \frac{\sqrt{1-e^2}\cos f}{ane}u_r - \frac{\sqrt{1-e^2}(2+e\cos f)\sin f}{ane(1+e\cos f)}u_t\right) \\
\\
&= \frac{n^2a(1+e\cos f)^2 e\cos f}{(1-e^2)^2} + \left(-\frac{e^2\sin^2 f}{1-e^2} + \frac{\sin^2 f}{1-e^2} + \cos^2 f\right)u_r \\
&+ \left(-\frac{(1+e\cos f)e\sin f}{1-e^2} + \frac{\left[(2+e\cos f)\cos f + e\right]\sin f}{(1-e^2)(1+e\cos f)} - \frac{(2+e\cos f)\sin f\cos f}{1+e\cos f}\right)u_t.
\end{aligned}$$

Where the terms in front of u_r can easily be simplified to one, the result of the terms in front of u_t is less obvious. Working out these terms carefully, however, one obtains that this term completely vanishes:

$$\begin{aligned}
\ddot{r}(u_t) &= -\frac{(1+e\cos f)e\sin f}{1-e^2} + \frac{\left[(2+e\cos f)\cos f + e\right]\sin f}{(1-e^2)(1+e\cos f)} - \frac{(2+e\cos f)\sin f\cos f}{1+e\cos f} \\
&= \frac{\left[(2+e\cos f)\cos f + e\right]\sin f - (1+e\cos f)^2 e\sin f - (1-e^2)(2+e\cos f)\sin f\cos f}{(1-e^2)(1+e\cos f)} \\
&= \frac{(2+e\cos f)\sin f\cos f - (2+e\cos f)e^2\sin f\cos f - (1-e^2)(2+e\cos f)\sin f\cos f}{(1-e^2)(1+e\cos f)} \\
&= 0,
\end{aligned}$$

such that the radial acceleration is as follows:

$$
\begin{aligned}
\ddot{r} &= \frac{n^2 a(1+e\cos f)^2 e\cos f}{(1-e^2)^2} + u_r \\
&= \frac{\mu(1+e\cos f)^2 e\cos f}{a^2(1-e^2)^2} + u_r \\
&= \frac{\mu}{r^2} e\cos f + u_r.
\end{aligned} \tag{A-15}
$$

Once again, a surprisingly easy result is obtained which adheres to the intuition one might have: it is zero for perfectly circular orbits and can be changed with the radial control input[2].

Utilising this result, the relative radial acceleration can be approximated as:

$$
\begin{aligned}
\delta\ddot{r} &= \ddot{r}_d - \ddot{r}_c \\
&= \frac{\mu}{r_d^2} e_d \cos f_d + u_r \\
&\approx \frac{\mu}{r_c^2} e_c \cos f_d - 2\frac{\mu}{r_c^2}\frac{e_c\cos f_d}{r_c}(r_d - r_c) + \frac{\mu}{r_c^2}\cos f_d(e_d - e_c) + u_r \\
&= \frac{\mu}{r_c^2}\delta e_x^f + u_r,
\end{aligned}
$$

which allows for the link between Eq. (A-11) and Eq. (A-14):

$$
\begin{aligned}
\delta\dot{r} = n_c r_c \delta e_y^f \quad \wedge \quad \delta\dot{e}_y^f = \frac{1}{a_c n_c} u_r + n_c \delta e_x^f \implies \delta\ddot{r} &= n_c r_c\left(\frac{1}{a_c n_c} u_r + n_c \delta e_x^f\right) \\
&= \frac{\mu}{r_c^2}\delta e_x^f + u_r.
\end{aligned}
$$

Thus, computing the radial acceleration directly yields the same result as combining the obtained formulas for δr and δe_y^f.

The second derivative of λ^f is found through the multiplication of its partial derivatives with respect to r, e and f and their derivatives, where $\dot{r}$ is obtained from Eq. (A-10) and the

[2] The control inputs are, as mentioned before, control accelerations instead of control forces. These could easily be swapped by dividing the control input by the satellite's mass.

others from Eq. (2-14):

$$
\begin{aligned}
\ddot{\lambda}^f &= -1\frac{1}{2}\frac{\dot{\lambda}^f}{r}\dot{r} + \frac{1}{2}\frac{\dot{\lambda}^f}{1+e\cos f}\cos f\dot{e} - \frac{1}{2}\frac{\dot{\lambda}^f}{1+e\cos f}e\sin f\dot{f} \\
&= -1\frac{1}{2}\frac{\dot{\lambda}^f}{r}\sqrt{\frac{\mu}{a(1-e^2)}}e\sin f \\
&+ \frac{1}{2}\frac{\dot{\lambda}^f}{1+e\cos f}\cos f\left(\frac{\sqrt{1-e^2}\sin f}{an}u_r + \frac{\sqrt{1-e^2}\left[(2+e\cos f)\cos f + e\right]}{an(1+e\cos f)}u_t\right) \\
&- \frac{1}{2}\frac{\dot{\lambda}^f}{1+e\cos f}e\sin f\left(\dot{\theta}_{\mathrm{f}} + \frac{\sqrt{1-e^2}\cos f}{ane}u_r - \frac{\sqrt{1-e^2}(2+e\cos f)\sin f}{ane(1+e\cos f)}u_t\right) \\
&= -2\frac{(\dot{\lambda}^f)^2}{1+e\cos f}e\sin f \\
&+ \frac{1}{2}\left(\frac{\dot{\lambda}^f\sqrt{1-e^2}\sin f\cos f}{an(1+e\cos f)} - \frac{\dot{\theta}_{\mathrm{f}}\sqrt{1-e^2}\sin f\cos f}{an(1+e\cos f)}\right)u_r \\
&+ \frac{1}{2}\left(\frac{\dot{\lambda}^f\sqrt{1-e^2}\left[(2+e\cos f)\cos f + e\right]\cos f}{an(1+e\cos f)^2} + \frac{\dot{\lambda}^f\sqrt{1-e^2}(2+e\cos f)\sin^2 f}{an(1+e\cos f)^2}\right)u_t.
\end{aligned}
$$

The term for u_t requires some work but yields a simple result once simplified:

$$
\begin{aligned}
\ddot{\lambda}^f(u_t) &= \frac{1}{2}\left(\frac{\dot{\lambda}^f\sqrt{1-e^2}\left[(2+e\cos f)\cos f + e\right]\cos f}{an(1+e\cos f)^2} + \frac{\dot{\lambda}^f\sqrt{1-e^2}(2+e\cos f)\sin^2 f}{an(1+e\cos f)^2}\right) \\
&= \frac{1}{2}\frac{\dot{\lambda}^f\sqrt{1-e^2}\left[(2+e\cos f) + e\cos f\right]}{an(1+e\cos f)^2} \\
&= \frac{\dot{\lambda}^f\sqrt{1-e^2}}{an(1+e\cos f)} \\
&= \sqrt{\frac{a(1-e^2)}{r^3(1+e\cos f)}} \\
&= \frac{1}{r},
\end{aligned}
$$

such that the final result is equal to:

$$
\ddot{\lambda}^f = -2\frac{\mu}{r^3}e\sin f + \frac{u_t}{r}. \tag{A-16}
$$

A sanity check confirms that this result can be correct, as the second derivative is zero for the reference orbit, and the control acceleration has to be divided by the current radius to obtain the angular acceleration. Using this result, the relative angular acceleration is equal to:

$$\begin{aligned}
\delta\ddot{\lambda}^f &= \ddot{\lambda}^f_d - \ddot{\lambda}^f_c \\
&= -2\frac{\mu}{r_d^3} e_d \sin f_d + \frac{u_t}{r_d} \\
&\approx -2\frac{\mu}{r_c^3} e_c \sin f_d + 6\frac{\mu}{r_c^3}\frac{e_c \sin f_d}{r_c}(r_d - r_c) - 2\frac{\mu}{r_c^3}\sin f_d(e_d - e_c) + \frac{u_t}{r_c} \\
&= -2\frac{\mu}{r_c^3}\delta e^f_y + \frac{u_t}{r_c},
\end{aligned}$$

which confirms that the approximation for $\delta\dot{e}^f_x$ is accurate:

$$\delta\dot{\lambda}^f = -\frac{3}{2}\frac{n_c}{r_c}\delta r + \frac{1}{2}n_c\delta e^f_x \;\wedge\; \delta\dot{e}^f_x = \frac{2}{a_c n_c}u_t - n_c\delta e^f_y \implies \begin{aligned}[t]
\delta\ddot{\lambda}^f &= -\frac{3}{2}\frac{n_c}{r_c}(n_c r_c \delta e^f_y) \\
&\quad + \frac{1}{2}n_c(\frac{2}{a_c n_c}u_t - n_c\delta e^f_y) \\
&= -2\frac{\mu}{r_c^3}\delta e^f_y + \frac{u_t}{r_c}.
\end{aligned}$$

Comparison With Hill-Clohessy-Wiltshire Model

It is possible to link these results to the absolute HCW model from Eq. (A-1). For the in-plane equations of motions, those results can be rewritten as:

$$\ddot{r} = r\dot{\varphi}^2 - n^2 r + u_r, \tag{A-17}$$

$$\ddot{\varphi} = -2\frac{\dot{r}}{r}\dot{\varphi} + \frac{u_t}{r}, \tag{A-18}$$

and recall the results for $\dot{\lambda}^f$ and $\dot{r}$ from Eqs. (A-6) and (A-10):

$$\begin{aligned}
\dot{\lambda}_f &= \sqrt{\frac{\mu(1 + e\cos f)}{r^3}} \\
\dot{r} &= \sqrt{\frac{\mu}{a(1 - e^2)}} e \sin f.
\end{aligned}$$

Using $\dot{\lambda}_f$ instead of $\dot{\varphi}$ in Eq. (A-17), one obtains for $\ddot{r}$:

$$\begin{aligned}
\ddot{r} &= \frac{\mu(1 + e\cos f)}{r^2} - \frac{\mu}{r^2} + u_r \\
&= \frac{\mu}{r^2} e\cos f + u_r,
\end{aligned}$$

where similar to the derivation in [23] a perfectly circular orbit is assumed, therefore r being equal to a. This result is equal to that obtained in Eq. (A-15).

The same can be shown for $\ddot{\varphi}$ in Eq. (A-18):

$$\begin{aligned}
\ddot{\varphi} &= -2\frac{e\sin f}{r}\sqrt{\frac{\mu^2(1 + e\cos f)}{a(1 - e^2)r^3}} + \frac{u_t}{r} \\
&= -2\frac{\mu}{r^3} e\sin f + \frac{u_t}{r},
\end{aligned}$$

which is equal to the result obtained in Eq. (A-16).

A-3-4 Out-Of-Plane States

The easiest variables to control the out-of-plane motion are simply $i_d - i_c$ and $\Omega_d - \Omega_c$ directly, where it is common to multiply $\Omega_d - \Omega_c$ with $\sin i_c$ such as in the ROE model in Eq. (2-16) to slightly simplify the results. This, however, results in a LTV model as the B matrix depends on either θ or υ. By choosing a different state variable, it is possible to create a LTI model.

One possible option is provided in [42], where they use the so-called equinoctial variables. Although the details of the equinoctial variables are not important, three variables are used for the out-of-plane control and are therefore relevant:

$$\begin{aligned}
\Gamma_1 &:= \theta + \Omega, \\
\Gamma_2 &:= \cos\Omega \tan\frac{i}{2}, \\
\Gamma_3 &:= \sin\Omega \tan\frac{i}{2}.
\end{aligned}$$

They define two relative states, $\delta\xi_x$ and $\delta\xi_y$, which make use of a reflection matrix with angle $\Gamma_{1,d}$:

$$\begin{aligned}
\begin{bmatrix}\delta\xi_x \\ \delta\xi_y\end{bmatrix} &= \begin{bmatrix}\cos\Gamma_{1,d} & \sin\Gamma_{1,d} \\ \sin\Gamma_{1,d} & -\cos\Gamma_{1,d}\end{bmatrix}\begin{bmatrix}\Gamma_{2,d} - \Gamma_{2,c} \\ \Gamma_{3,d} - \Gamma_{3,c}\end{bmatrix} \\
&= \begin{bmatrix}\cos\Gamma_{1,d} & \sin\Gamma_{1,d} \\ \sin\Gamma_{1,d} & -\cos\Gamma_{1,d}\end{bmatrix}\begin{bmatrix}\Gamma_{2,d} \\ \Gamma_{3,d}\end{bmatrix} - \begin{bmatrix}\cos\Gamma_{1,d} & \sin\Gamma_{1,d} \\ \sin\Gamma_{1,d} & -\cos\Gamma_{1,d}\end{bmatrix}\begin{bmatrix}\Gamma_{2,c} \\ \Gamma_{3,c}\end{bmatrix}.
\end{aligned}$$

Working out the first term yields:

$$\begin{bmatrix}\cos\Gamma_{1,d} & \sin\Gamma_{1,d} \\ \sin\Gamma_{1,d} & -\cos\Gamma_{1,d}\end{bmatrix}\begin{bmatrix}\Gamma_{2,d} \\ \Gamma_{3,d}\end{bmatrix} = \begin{bmatrix}\cos(\theta_d + \Omega_d)\cos\Omega_d \tan\frac{i_d}{2} + \sin(\theta_d + \Omega_d)\sin\Omega_d \tan\frac{i_d}{2} \\ \sin(\theta_d + \Omega_d)\cos\Omega_d \tan\frac{i_d}{2} - \cos(\theta_d + \Omega_d)\sin\Omega_d \tan\frac{i_d}{2}\end{bmatrix},$$

which, using the following four trigonometric identities:

$$\begin{aligned}
\cos\alpha\cos\beta &= \frac{1}{2}(\cos(\alpha - \beta) + \cos(\alpha + \beta)), \\
\sin\alpha\sin\beta &= \frac{1}{2}(\cos(\alpha - \beta) - \cos(\alpha + \beta)), \\
\sin\alpha\cos\beta &= \frac{1}{2}(\sin(\alpha + \beta) + \sin(\alpha - \beta)), \\
\cos\alpha\sin\beta &= \frac{1}{2}(\sin(\alpha + \beta) - \sin(\alpha - \beta)),
\end{aligned}$$

can be rewritten to:

$$\begin{aligned}
\begin{bmatrix}\cos\Gamma_{1,d} & \sin\Gamma_{1,d} \\ \sin\Gamma_{1,d} & -\cos\Gamma_{1,d}\end{bmatrix}\begin{bmatrix}\Gamma_{2,d} \\ \Gamma_{3,d}\end{bmatrix} &= \frac{1}{2}\tan\frac{i_d}{2}\begin{bmatrix}\cos\theta_d + \cos(\theta_d + 2\Omega_d) + \cos\theta_d - \cos(\theta_d + 2\Omega_d) \\ \sin(\theta_d + 2\Omega_d) + \sin\theta_d - \sin(\theta_d + 2\Omega_d) + \sin\theta_d\end{bmatrix} \\
&= \tan\frac{i_d}{2}\begin{bmatrix}\cos\theta_d \\ \sin\theta_d\end{bmatrix}.
\end{aligned}$$

A similar approach can be followed for the second term, where not as many terms cancel out:

$$\begin{aligned}
\begin{bmatrix}\cos\Gamma_{1,d} & \sin\Gamma_{1,d} \\ \sin\Gamma_{1,d} & -\cos\Gamma_{1,d}\end{bmatrix}\begin{bmatrix}\Gamma_{2,c} \\ \Gamma_{3,c}\end{bmatrix} &= \begin{bmatrix}\cos(\theta_d + \Omega_d)\cos\Omega_c \tan\frac{i_c}{2} + \sin(\theta_d + \Omega_d)\sin\Omega_c \tan\frac{i_c}{2} \\ \sin(\theta_d + \Omega_d)\cos\Omega_c \tan\frac{i_c}{2} - \cos(\theta_d + \Omega_d)\sin\Omega_c \tan\frac{i_c}{2}\end{bmatrix} \\
&= \tan\frac{i_c}{2}\begin{bmatrix}\cos(\theta_d + \Omega_d - \Omega_c) \\ \sin(\theta_d + \Omega_d - \Omega_c)\end{bmatrix},
\end{aligned}$$

such that the results for $\delta\xi_x$ and $\delta\xi_y$ can be obtained:

$$\delta\xi_x = \cos\theta_d \tan\frac{i_d}{2} - \cos(\theta_d + \Omega_d - \Omega_c)\tan\frac{i_c}{2}$$
$$\delta\xi_y = \sin\theta_d \tan\frac{i_d}{2} - \sin(\theta_d + \Omega_d - \Omega_c)\tan\frac{i_c}{2}.$$

These terms look rather peculiar, but their function quickly becomes clear when looking at their derivatives. Without any assumptions, the derivative of ξ_x remains complex and nonlinear:

$$\begin{aligned}
\delta\dot{\xi}_x &= \frac{d}{dt}\left[\cos\theta_d \tan\frac{i_d}{2} - \cos(\theta_d + \Omega_d - \Omega_c)\tan\frac{i_c}{2}\right] \\
&= -\dot{\theta}_d \sin\theta_d \tan\frac{i_d}{2} + \dot{\theta}_d \sin(\theta_d + \Omega_d - \Omega_c)\tan\frac{i_c}{2} + \frac{1}{2}\frac{\cos\theta_d}{\cos^2\frac{i_d}{2}}\frac{d\,i_d}{dt} + \dot{\Omega}_d \sin(\theta_d + \Omega_d - \Omega_c)\tan\frac{i_c}{2} \\
&= -\dot{\theta}_d\,\delta\xi_y + \frac{1}{2}\frac{\sqrt{1-e_d^2}\cos^2\theta_d}{a_d n_d(1+e_d\cos f_d)\cos^2\frac{i_d}{2}}u_n + \frac{\sqrt{1-e^2}\sin\theta_d \sin(\theta_d+\Omega_d-\Omega_c)\tan\frac{i_c}{2}}{a_d n_d(1+e_d\cos f_d)\sin i_d}u_n.
\end{aligned}$$

However, if it is assumed that $\Omega_d - \Omega_c$ and $i_d - i_c$ are both relatively small, the following trigonometric identities can be used:

$$\cos^2\frac{\alpha}{2} = \frac{1}{2}(1+\cos\alpha),$$
$$\tan\frac{\alpha}{2} = \frac{\sin\alpha}{1+\cos\alpha},$$

to find the following approximation of $\delta\dot{\xi}_x$:

$$\begin{aligned}
\delta\dot{\xi}_x &= -\dot{\theta}_d\,\delta\xi_y + \frac{\sqrt{1-e_d^2}\cos^2\theta_d u_n}{a_d n_d(1+e_d\cos f_d)(1+\cos i_d)} + \frac{\sqrt{1-e^2}\sin\theta_d\sin(\theta_d+\Omega_d-\Omega_c)\sin i_c u_n}{a_d n_d(1+e_d\cos f_d)\sin i_d(1+\cos i_c)} \\
&\approx -\dot{\theta}_d\,\delta\xi_y + \frac{\sqrt{1-e_d^2}\cos^2\theta_d u_n}{a_d n_d(1+e_d\cos f_d)(1+\cos i_d)} + \frac{\sqrt{1-e^2}\sin^2\theta_d u_n}{a_d n_d(1+e_d\cos f_d)(1+\cos i_d)} \\
&= -\dot{\theta}_d\,\delta\xi_y + \frac{\sqrt{1-e_d^2}}{a_d n_d(1+e_d\cos f_d)(1+\cos i_d)}u_n.
\end{aligned}$$

This result is still nonlinear and time-varying, but when assuming a circular orbit and that $\dot{\theta}_d \approx n_d$, $n_d \approx n_c$ and $a_d \approx a_c$, the LTI form is obtained:

$$\delta\dot{\xi}_x = -n_c\,\delta\xi_y + \frac{1}{a_c n_c(1+\cos i_c)}u_n.$$

Following the same steps, the derivative of $\delta\xi_y$ can be obtained:

$$\begin{aligned}
\delta\dot{\xi}_y &= \frac{d}{dt}\left[\sin\theta_d \tan\frac{i_d}{2} - \sin(\theta_d + \Omega_d - \Omega_c)\tan\frac{i_c}{2}\right] \\
&= \dot{\theta}_d \cos\theta_d \tan\frac{i_d}{2} - \dot{\theta}_d \cos(\theta_d + \Omega_d - \Omega_c)\tan\frac{i_c}{2} + \frac{1}{2}\frac{\sin\theta_d}{\cos^2\frac{i_d}{2}}\frac{d\,i}{dt} + \dot{\Omega}_d \cos(\theta_d + \Omega_d - \Omega_c)\tan\frac{i_c}{2} \\
&= \dot{\theta}_d\delta\xi_x + \frac{1}{2}\frac{\sqrt{1-e_d^2}\sin\theta_d\cos\theta_d}{a_d n_d(1+e_d\cos f_d)\cos^2\frac{i_d}{2}}u_n + \frac{\sqrt{1-e^2}\sin\theta_d\cos(\theta_d+\Omega_d-\Omega_c)\tan\frac{i_c}{2}}{a_d n_d(1+e_d\cos f_d)\sin i_d}u_n \\
&\approx n_c\,\delta\xi_x.
\end{aligned}$$

A-3-5 J_2 Perturbations With Blend Model

To add the effect of J_2 perturbations to the model, first some basic equations are presented. This is followed by the derivation of the effect of J_2 perturbations on the different states, similar to the structure as before. First, the radial state is presented, followed by the angular state, the eccentric states and finally the out-of-plane states.

Basic Equations

Terms to correct for J_2 perturbations for mean orbital elements are almost exclusively given for $\dot{M}$, $\dot{\omega}$ and $\dot{\Omega}$. However, as the goal is to use the argument of latitude and it was shown before that there can be a large difference between $\dot{f}+\dot{\omega}$ and $\dot{M}+\dot{\omega}$ (see Section A-3-1), this poses a challenge at first. Luckily, upon closer inspection, the problem resolves itself. Recall from Eq. (2-5), that, when disregarding all terms of $\mathcal{O}(e^2)$ or smaller:

$$M = f - 2e\sin f,$$

which, after differentiating, yields:

$$\begin{aligned}
\dot{M} &= \dot{f} - 2\dot{e}\sin f - 2e\dot{f}\cos f \\
&= \dot{f} - 2\left(\frac{\sqrt{1-e^2}\sin^2 f}{an}u_r + \frac{\sqrt{1-e^2}\left[(2+e\cos f)\sin f\cos f + e\sin f\right]}{an(1+e\cos f)}u_t\right) \\
&- 2\left(\frac{ne(1+e\cos f)^2\cos f}{(1-e^2)^{\frac{3}{2}}} + \frac{\sqrt{1-e^2}\cos^2 f}{an}u_r - \frac{\sqrt{1-e^2}(2+e\cos f)\sin f\cos f}{an(1+e\cos f)}u_t\right) \\
&\approx \dot{f} - \frac{2}{an}u_r.
\end{aligned}$$

Note that this is also the difference obtained between Eq. (A-4) and Eq. (A-6) when the eccentricity is set to zero in the latter, and that this difference comes from the control inputs as opposed to the Keplerian gravitational forces. Then, if $\dot{M}$ would be split into three parts: Keplerian forces, J_2 perturbations and control inputs:

$$\dot{M}_{\text{Kepler}} + \dot{M}_{\text{J}_2} + \dot{M}_{\text{Control}} = \dot{f}_{\text{Kepler}} + \dot{f}_{\text{J}_2} + \dot{f}_{\text{Control}} - \frac{2}{an}u_r,$$

combined with:

$$\begin{aligned}
\dot{M}_{\text{Control}} &= \dot{f}_{\text{Control}} - \frac{2}{an}u_r, \\
\dot{M}_{\text{Kepler}} &= \dot{f}_{\text{Kepler}} = n,
\end{aligned}$$

allows for the following conclusion:

$$\dot{M}_{\text{J}_2} = \dot{f}_{\text{J}_2}.$$

Thus, it can be assumed that the effect of the J_2 perturbations on the true anomaly is equal to that of the effect on the mean anomaly. This makes it relatively easy to add corrections for J_2 perturbations to the model. To do so, firstly a group of constant parameters are denoted by γ:

$$\gamma := \frac{3}{4}J_2R_e^2\sqrt{\mu},$$

such that the J_2 perturbations can now be expressed as:

$$\begin{bmatrix} \dot{f}_{\mathrm{J}_2} \\ \dot{\omega}_{\mathrm{J}_2} \\ \dot{\Omega}_{\mathrm{J}_2} \end{bmatrix} = \frac{\gamma}{a^{\frac{7}{2}}(1-e^2)^2} \begin{bmatrix} \eta(3\cos^2 i - 1) \\ 5\cos^2 i - 1 \\ -2\cos i \end{bmatrix} = \frac{\gamma(1-e^2)^{\frac{3}{2}}}{r^{\frac{7}{2}}(1+e\cos f)^{\frac{7}{2}}} \begin{bmatrix} \eta(3\cos^2 i - 1) \\ 5\cos^2 i - 1 \\ -2\cos i \end{bmatrix}. \tag{A-19}$$

Radial State

When looking at the effect of J_2 perturbations on δr, note how in Eq. (A-9) only the last term is affected by the J_2 perturbations. That means that:

$$\begin{aligned} \dot{r}_{\mathrm{J}_2} &= \frac{ae(1-e^2)\sin f}{(1+e\cos f)^2}\dot{f}_{\mathrm{J}_2} \\ &= \frac{ae(1-e^2)\sin f}{(1+e\cos f)^2}\frac{\gamma(1-e^2)^{\frac{3}{2}}}{r^{\frac{7}{2}}(1+e\cos f)^{\frac{7}{2}}}\eta(3\cos^2 i - 1) \\ &= \gamma\frac{e(1-e^2)^2\sin f}{(1+e\cos f)^{\frac{9}{2}}r^{\frac{5}{2}}}(3\cos^2 i - 1). \end{aligned}$$

The relative dynamics are found with a first-order Taylor expansion around (r_c, e_c, f_d, i_c), which has the following property:

$$\begin{aligned} \dot{r}_{\mathrm{J}_2,\mathrm{Taylor}} &= \gamma\frac{e_c(1-e_c^2)^3\sin f_d}{(1+e_c\cos f_d)^{\frac{9}{2}}r_c^{\frac{5}{2}}}(3\cos^2 i_c - 1) \\ &= \dot{r}_{\mathrm{J}_2,c}, \end{aligned}$$

because e_c is assumed to be zero. The Taylor expansion then yields:

$$\begin{aligned} \delta\dot{r}_{\mathrm{J}_2} &= \dot{r}_{\mathrm{J}_2,d} - \dot{r}_{\mathrm{J}_2,c} \\ &\approx \dot{r}_{\mathrm{J}_2,\mathrm{Taylor}} - \frac{5}{2}\dot{r}_{\mathrm{J}_2,\mathrm{Taylor}}\frac{r_d - r_c}{r_c} + \left(\frac{1}{e_c} - 6\frac{e_c}{1-e_c^2} - \frac{9}{2}\frac{\cos f_d}{1+e_c\cos f_d}\right)\dot{r}_{\mathrm{J}_2,\mathrm{Taylor}}(e_d - e_c) \\ &+ \frac{6\cos i_c\sin i_c}{3\cos^2 i - 1}\dot{r}_{\mathrm{J}_2,\mathrm{Taylor}}(i_d - i_c) - \dot{r}_{\mathrm{J}_2,c} \\ &= \gamma\, r_c^{-\frac{5}{2}}(3\cos^2 i_c - 1)\delta e_y^f, \end{aligned}$$

where almost all terms drop out in the last step as e_c is zero.

Angular State

The angular state λ^f is a summation of f, ω and $\Omega\cos i$ as seen in Eq. (A-7). These terms can individually be analysed to find the total effect of them combined.

Firstly the effect of J_2 perturbations on the true anomaly is analysed. Through a first-order Taylor expansion around (r_c, e_c, f_d, i_c), the following term is obtained:

$$\dot{f}_{\mathrm{J}_2,\mathrm{Taylor}} = \frac{\gamma(1-e_c^2)^2}{r_c^{\frac{7}{2}}(1+e_c\cos f_d)^{\frac{7}{2}}}(3\cos^2 i_c - 1),$$

which is again equal to $\dot{f}_{\mathrm{J}_2,c}$ as e_c is zero. Then the relative dynamics can be found:

$$\begin{aligned}
\delta\dot{f}_{\mathrm{J}_2} &= \dot{f}_{\mathrm{J}_2,d} - \dot{f}_{\mathrm{J}_2,c} \\
&= \frac{\gamma(1-e_d^2)^2}{r_d^{\frac{7}{2}}(1+e_d\cos f_d)^{\frac{7}{2}}}(3\cos^2 i_d - 1) - \dot{f}_{\mathrm{J}_2,c} \\
&\approx \dot{f}_{\mathrm{J}_2,\mathrm{Taylor}} - \frac{7}{2}\frac{\dot{f}_{\mathrm{J}_2,\mathrm{Taylor}}}{r_c}(r_d - r_c) - \left(4\frac{e_c}{1-e_c^2} + \frac{7}{2}\frac{\cos f_d}{1+e_c\cos f_d}\right)\dot{f}_{\mathrm{J}_2,\mathrm{Taylor}}(e_d - e_c) \\
&\quad - 6\frac{\sin i_c \cos i_c \dot{f}_{\mathrm{J}_2,\mathrm{c}}}{3\cos^2 i_c - 1}(i_d - i_c) - \dot{f}_{\mathrm{J}_2,\mathrm{c}} \\
&= -\frac{7}{2}\frac{\dot{f}_{\mathrm{J}_2,\mathrm{c}}}{r_c}\delta r - \frac{7}{2}\dot{f}_{\mathrm{J}_2,\mathrm{c}}\delta e_x^f - 6\frac{\sin i_c \cos i_c \dot{f}_{\mathrm{J}_2,\mathrm{Taylor}}}{3\cos^2 i_c - 1}(i_d - i_c) \\
&= -\frac{7}{2}\gamma\frac{3\cos^2 i_c - 1}{r_c^{\frac{9}{2}}}\delta r - \frac{7}{2}\gamma\frac{3\cos^2 i_c - 1}{r_c^{\frac{7}{2}}}\delta e_x^f.
\end{aligned} \tag{A-20}$$

Here, the last term can be discarded in the second-to-last line as it is assumed that the inclinations are close to each other.

A similar process can be followed for the argument of periapsis, where the linearisation yields the following term:

$$\dot{\omega}_{\mathrm{J}_2,\mathrm{Taylor}} = \frac{\gamma(1-e_c^2)^{\frac{3}{2}}}{r_c^{\frac{7}{2}}(1+e_c\cos f_d)^{\frac{7}{2}}}(5\cos^2 i_c - 1).$$

This yields the following formula for the satellite after a Taylor expansion again around (r_c, e_c, f_d, i_c):

$$\begin{aligned}
\delta\dot{\omega}_{\mathrm{J}_2} &= \dot{\omega}_{\mathrm{J}_2,d} - \dot{\omega}_{\mathrm{J}_2,c} \\
&= \frac{\gamma(1-e_d^2)^{\frac{3}{2}}}{r_d^{\frac{7}{2}}(1+e_d\cos f_d)^{\frac{7}{2}}}(5\cos^2 i_d - 1) - \dot{\omega}_{\mathrm{J}_2,c} \\
&\approx \dot{\omega}_{\mathrm{J}_2,\mathrm{Taylor}} - \frac{7}{2}\frac{\dot{\omega}_{\mathrm{J}_2,\mathrm{Taylor}}}{r_c}(r_d - r_c) - \left(\frac{3e_c\,\dot{\omega}_{\mathrm{J}_2,\mathrm{Taylor}}}{1-e_c^2} + \frac{7}{2}\frac{\dot{\omega}_{\mathrm{J}_2,\mathrm{Taylor}}\cos f_d}{1+e_c\cos f_d}\right)(e_d - e_c) \\
&\quad - 10\frac{\sin i_c \cos i_c \dot{\omega}_{\mathrm{J}_2,\mathrm{Taylor}}}{5\cos^2 i_c - 1}(i_d - i_c) - \dot{\omega}_{\mathrm{J}_2,c} \\
&= -\frac{7}{2}\frac{\dot{\omega}_{\mathrm{J}_2,\mathrm{c}}}{r_c}\delta r - \frac{7}{2}\dot{\omega}_{\mathrm{J}_2,\mathrm{Taylor}}\delta e_x^f - 10\frac{\sin i_c \cos i_c \dot{\omega}_{\mathrm{J}_2,\mathrm{Taylor}}}{5\cos^2 i_c - 1}(i_d - i_c) \\
&= -\frac{7}{2}\gamma\frac{5\cos^2 i_c - 1}{r_c^{\frac{9}{2}}}\delta r - \frac{7}{2}\gamma\frac{5\cos^2 i_c - 1}{r_c^{\frac{7}{2}}}\delta e_x^f.
\end{aligned} \tag{A-21}$$

Finally, the same procedure can be followed for the RAAN:

$$\dot{\Omega}_{\mathrm{J}_2,\mathrm{Taylor}} = -2\frac{\gamma(1-e_c^2)^{\frac{3}{2}}}{r_c^{\frac{7}{2}}(1+e_c\cos f_d)^{\frac{7}{2}}}\cos i_c,$$

with the following result after the first-order Taylor expansion:

$$
\begin{aligned}
\delta\dot{\Omega}_{\mathrm{J}_2} &= \dot{\Omega}_{\mathrm{J}_2,d} - \dot{\Omega}_{\mathrm{J}_2,c} \\
&- 2\frac{\gamma(1-e_d^2)^{\frac{3}{2}}}{r_d^{\frac{7}{2}}(1+e_d\cos f_d)^{\frac{7}{2}}}\cos i_d - \dot{\Omega}_{\mathrm{J}_2,c} \\
&\approx \dot{\Omega}_{\mathrm{J}_2,\mathrm{Taylor}} - \frac{7}{2}\frac{\dot{\Omega}_{\mathrm{J}_2,\mathrm{Taylor}}}{r_c}(r_d - r_c) - \left(3\frac{e_c\dot{\Omega}_{\mathrm{J}_2,\mathrm{Taylor}}}{1-e_c^2} + \frac{7}{2}\frac{\dot{\Omega}_{\mathrm{J}_2,\mathrm{Taylor}}\cos f_d}{1+e_c\cos f_d}\right)(e_d - e_c) \\
&- \tan i_c\dot{\Omega}_{\mathrm{J}_2,\mathrm{Taylor}}(i_d - i_c) - \dot{\Omega}_{\mathrm{J}_2,c} \\
&= -\frac{7}{2}\frac{\dot{\Omega}_{\mathrm{J}_2,\mathrm{Taylor}}}{r_c}\delta r - \frac{7}{2}\dot{\Omega}_{\mathrm{J}_2,\mathrm{Taylor}}\delta e_x^f - \tan i_c\dot{\Omega}_{\mathrm{J}_2,\mathrm{c}}(i_d - i_c) \\
&= 7\gamma\, r_c^{-\frac{9}{2}}\cos i_c\delta r + 7\gamma\, r_c^{-\frac{7}{2}}\cos i_c\delta e_x^f. \qquad \text{(A-22)}
\end{aligned}
$$

Using the following relationship:

$$
\begin{aligned}
\delta\dot{\lambda}^f &= \dot{\lambda}_d^f - \dot{\lambda}_c^f \\
&= \dot{f}_d - \dot{f}_c + \dot{\omega}_d - \dot{\omega}_c + \left(\dot{\Omega}_d - \dot{\Omega}_c\right)\cos i_c \\
&= \delta\dot{f} + \delta\dot{\omega} + \delta\Omega\,\cos i_c,
\end{aligned}
$$

and using Eqs. (A-20) to (A-22), the effect of J_2 perturbation on $\dot{\lambda}^f$ can be found:

$$
\begin{aligned}
\delta\dot{\lambda}_{\mathrm{J}_2}^f &= \delta\dot{f}_{\mathrm{J}_2} + \delta\dot{\omega}_{\mathrm{J}_2} + \delta\Omega_{\mathrm{J}_2}\,\cos i_c \\
&= -\frac{7}{2}\gamma\frac{6\cos^2 i_c - 2}{r_c^{\frac{9}{2}}}\delta r - \frac{7}{2}\gamma\frac{6\cos^2 i_c - 2}{r_c^{\frac{7}{2}}}\delta e_x^f.
\end{aligned}
$$

Eccentric States

The eccentric states δe_x^f and δe_y^f are influenced by J_2 perturbations through their dependence on the true anomaly, which when analysing the derivatives yields:

$$
\begin{aligned}
\dot{e}_{x,\mathrm{J}_2}^f &= \dot{e}_{\mathrm{J}_2}\cos f - e\dot{f}_{\mathrm{J}_2}\sin f = e\dot{f}_{\mathrm{J}_2}\sin f, \\
\dot{e}_{y,\mathrm{J}_2}^f &= \dot{e}_{\mathrm{J}_2}\sin f + e\dot{f}_{\mathrm{J}_2}\cos f = e\dot{f}_{\mathrm{J}_2}\cos f.
\end{aligned}
$$

Similarly to before, two terms around which the Taylor expansion will be done are defined:

$$
\begin{aligned}
\dot{e}_{x,\mathrm{J}_2,\mathrm{Taylor}}^f &= -\gamma\, e_c\sin f_d\frac{(1-e_c^2)^2(3\cos^2 i_c - 1)}{r_c^{\frac{7}{2}}(1+e_c\cos f_d)^{\frac{7}{2}}}, \\
\dot{e}_{y,\mathrm{J}_2,\mathrm{Taylor}}^f &= \gamma\, e_c\cos f_d\frac{(1-e_c^2)^2(3\cos^2 i_c - 1)}{r_c^{\frac{7}{2}}(1+e_c\cos f_d)^{\frac{7}{2}}}.
\end{aligned}
$$

The first eccentric state δe_x^f, when analysing the relative dynamics with a Taylor expansion, yields:

$$
\begin{aligned}
\delta\dot{e}^f_{x,\mathrm{J}_2} &= -\gamma\, e_d \sin f_d \frac{(1-e_d^2)^2(3\cos^2 i_d - 1)}{r_d^{\frac{7}{2}}(1+e_d\cos f_d)^{\frac{7}{2}}} + \gamma\, e_c \sin f_c \frac{(1-e_c^2)^2(3\cos^2 i_c - 1)}{r_c^{\frac{7}{2}}(1+e_c\cos f_c)^{\frac{7}{2}}} \\
&\approx \dot{e}^f_{x,\mathrm{J}_2,\mathrm{Taylor}} - \frac{7}{2}\frac{\dot{e}^f_{x,\mathrm{J}_2,\mathrm{Taylor}}}{r_c}(r_d - r_c) - 6\frac{\sin i_c \cos i_c\, \dot{e}^f_{x,\mathrm{J}_2,\mathrm{Taylor}}}{3\cos^2 i_c - 1}(i_d - i_c) \\
&+ \left(\frac{1}{e_c} - 4\frac{e_c}{1-e_c^2} - \frac{7}{2}\frac{\cos f_d}{1+e_c\cos f_d}\right)\dot{e}^f_{x,\mathrm{J}_2,\mathrm{Taylor}}(e_d - e_c) - \dot{e}^f_{x,\mathrm{J}_2,c} \\
&= -\gamma \sin f_d \frac{(1-e_c^2)^2(3\cos^2 i_c - 1)}{r_c^{\frac{7}{2}}(1+e_c\cos f_d)^{\frac{7}{2}}}(e_d - e_c) \\
&= -\gamma\frac{3\cos^2 i_c - 1}{r_c^{\frac{7}{2}}}\delta e^f_y.
\end{aligned}
$$

A similar derivation gives the result for δe^f_y:

$$
\begin{aligned}
\delta\dot{e}^f_{y,\mathrm{J}_2} &= \gamma\, e_d \cos f_d \frac{(1-e_d^2)^2(3\cos^2 i_d - 1)}{r_d^{\frac{7}{2}}(1+e_d\cos f_d)^{\frac{7}{2}}} - \gamma\, e_c \cos f_c \frac{(1-e_c^2)^2(3\cos^2 i_c - 1)}{r_c^{\frac{7}{2}}(1+e_c\cos f_c)^{\frac{7}{2}}} \\
&\approx \dot{e}^f_{y,\mathrm{J}_2,\mathrm{Taylor}} - \frac{7}{2}\frac{\dot{e}^f_{y,\mathrm{J}_2,\mathrm{Taylor}}}{r_c}(r_d - r_c) - 6\frac{\sin i_c \cos i_c\, \dot{e}^f_{y,\mathrm{J}_2,\mathrm{Taylor}}}{3\cos^2 i_c - 1}(i_d - i_c) \\
&+ \left(\frac{1}{e_c} - 4\frac{e_c}{1-e_c^2} - \frac{7}{2}\frac{\cos f_d}{1+e_c\cos f_d}\right)\dot{e}^f_{y,\mathrm{J}_2,\mathrm{Taylor}}(e_d - e_c) - \dot{e}^f_{y,\mathrm{J}_2,c} \\
&= \gamma \cos f_d \frac{(1-e_c^2)^2(3\cos^2 i_c - 1)}{r_c^{\frac{7}{2}}(1+e_c\cos f_d)^{\frac{7}{2}}}(e_d - e_c) \\
&= \gamma\frac{3\cos^2 i_c - 1}{r_c^{\frac{7}{2}}}\delta e^f_x.
\end{aligned}
$$

Out-Of-Plane States

The derivation for the out-of-plane states $\delta\xi_x$ and $\delta\xi_y$ is complex. Recall for example the derivative from Section A-3-4:

$$
\begin{aligned}
\delta\dot{\xi}_{x,\mathrm{J}_2} &= \frac{d}{dt}\left[\cos\theta_d \tan\frac{i_d}{2} - \cos(\theta_d + \Omega_d - \Omega_c)\tan\frac{i_c}{2}\right]_{\mathrm{J}_2} \\
&= -\dot{\theta}_{\mathrm{J}_2,d}\,\delta\xi_y + \frac{1}{2}\frac{\cos\theta_d}{\cos^2\frac{i_d}{2}}\left[\frac{d\,i_d}{dt}\right]_{\mathrm{J}_2} + \dot{\Omega}_{\mathrm{J}_2,d}\sin(\theta_d + \Omega_d - \Omega_c)\tan\frac{i_c}{2},
\end{aligned}
$$

which shows that J_2 perturbations will have an effect on both the first and third terms (as $\frac{d\,i}{dt}$ is unaffected). The first term can be approximated as:

$$
\begin{aligned}
\dot{\theta}_{\mathrm{J}_2,d}\delta\xi_y &\approx \dot{\theta}_{\mathrm{J}_2,c}\delta\xi_y \\
&= (\dot{f}_{\mathrm{J}_2,c} + \dot{\omega}_{\mathrm{J}_2,c})\delta\xi_y \\
&= \gamma r_c^{-\frac{7}{2}}(8\cos^2 i_c - 2)\delta\xi_y,
\end{aligned}
$$

which is the desired LTI result. This is not the case for the third term, however:

$$\begin{aligned}\dot{\Omega}_{\mathrm{J}_2,d}\sin(\theta_d+\Omega_d-\Omega_c)\tan\frac{i_c}{2} &\approx \dot{\Omega}_{\mathrm{J}_2,c}\sin(\theta_d+\Omega_d-\Omega_c)\tan\frac{i_c}{2}\\ &= -2\gamma r_c^{-\frac{7}{2}}\cos i_c\sin(\theta_d+\Omega_d-\Omega_c)\tan\frac{i_c}{2}\\ &\approx -2\gamma r_c^{-\frac{7}{2}}\cos i_c\sin\theta_d\tan\frac{i_c}{2},\end{aligned}$$

which will always be time-varying due to the dependency on the argument of latitude. However, because the other terms are almost constant, the average of this term is zero and leaving this term out therefore produces no long-term errors. Thus, the total effect of J_2 perturbations on $\delta\xi_x$ can be modelled as:

$$\delta\dot{\xi}_{x,\mathrm{J}_2} = -\gamma r_c^{-\frac{7}{2}}(8\cos^2 i_c - 2)\delta\xi_y.$$

The approach for $\delta\xi_y$ is comparable:

$$\begin{aligned}\delta\dot{\xi}_{y,\mathrm{J}_2} &= \frac{d}{dt}\left[\sin\theta_d\tan\frac{i_d}{2} - \sin(\theta_d+\Omega_d-\Omega_c)\tan\frac{i_c}{2}\right]\\ &= \dot{\theta}_{\mathrm{J}_2,d}\delta\xi_x + \frac{1}{2}\frac{\sin\theta_d}{\cos^2\frac{i_d}{2}}\left[\frac{d\,i_d}{dt}\right]_{\mathrm{J}_2} + \dot{\Omega}_{\mathrm{J}_2,d}\cos(\theta_d+\Omega_d-\Omega_c)\tan\frac{i_c}{2}\\ &\approx \dot{\theta}_{\mathrm{J}_2,c}\delta\xi_x + \dot{\Omega}_{\mathrm{J}_2,c}\cos\theta_d\tan\frac{i_c}{2}\\ &\approx \gamma r_c^{-\frac{7}{2}}(8\cos^2 i_c - 2)\delta\xi_x.\end{aligned}$$

Appendix B

Control Parameters

A MPC controls the satellites in the different simulations. The following quadratic problem is solved in each iteration:

$$\begin{aligned}
\underset{\substack{x_0,\dots,x_T,\\ u_0,\dots,u_{T-1}}}{\text{minimize}} \quad & \sum_{t=0}^{T-1} \left[(x_t - x_r)^\mathsf{T} Q \, (x_t - x_r) + u_t^\mathsf{T} R \, u_t \right] + (x_T - x_r)^\mathsf{T} Q_T \, (x_T - x_r) \\
\text{subject to} \quad & x_{t+1} = A_t \, x_t + B_t \, u_t \quad \forall t \in \mathbb{Z}_0^{T-1}, \\
& x_{\min} \leq x_t \leq x_{\max} \qquad \forall t \in \mathbb{Z}_1^{T}, \\
& u_{\min} \leq u_t \leq u_{\max} \qquad \forall t \in \mathbb{Z}_0^{T-1}.
\end{aligned}$$

where the exact parameters can differ between different simulations and for different models. The parameters for the results from Chapter 2 are provided in Appendix B-1, after which the parameters from Chapter 3 are provided in Appendix B-2.

B-1 Model Comparisons

The different parameters used for the HCW, the ROE and the Blend model are shown in Sections B-1-1 to B-1-3, respectively.

B-1-1 Cylindrical Hill-Clohessy-Wiltshire Parameters

The control parameters used for the HCW model during the model comparisons are shown in Table B-1.

Table B-1: Control parameters for Hill-Clohessy-Wiltshire model during model comparisons.

Parameter	Description	Value
Q	State cost	$\texttt{diag}(25, 2500, 10^{-4}, 2500, 10^{4}, 10^{-4})$
Q_T	Terminal state cost	Q
R	Input cost	$\texttt{diag}(10^{-4}, 10^{-4}, 10^{-4})$
$x_{\max}$	Maximum state values	$\begin{bmatrix} 0.1 & 10 & 4 & 0.1 & \frac{1}{10}n_c & 0.1 \end{bmatrix}^\mathsf{T}$
$x_{\min}$	Minimum state values	$-x_{\max}$
$u_{\max}$	Maximum input values	$\begin{bmatrix} 0.1 & 0.1 & 0.1 \end{bmatrix}^\mathsf{T}$
$u_{\min}$	Minimum input values	$-u_{\max}$
T	Prediction horizon	20

B-1-2 Quasi-Nonsingular Relative Orbital Elements Parameters

The control parameters used for the ROE model during the model comparisons are shown in Table B-2.

Table B-2: Control parameters for Relative Orbital Elements model during model comparisons.

Parameter	Description	Value
Q	State cost	$\texttt{diag}(25r_c^2, 2500, 4\cdot 10^4, 4\cdot 10^4, 225, 225)$
Q_T	Terminal state cost	Q
R	Input cost	$\texttt{diag}(10^{-4}, 10^{-4}, 10^{-4})$
$x_{\max}$	Maximum state values	$\begin{bmatrix}\frac{0.1}{r_c} & 100 & 0.002 & 0.002 & 0.05 & 1\end{bmatrix}^\mathsf{T}$
$x_{\min}$	Minimum state values	$-x_{\max}$
$u_{\max}$	Maximum input values	$\begin{bmatrix}0.1 & 0.1 & 0.1\end{bmatrix}^\mathsf{T}$
$u_{\min}$	Minimum input values	$-u_{\max}$
T	Prediction horizon	20

B-1-3 Blend Parameters

The control parameters used for the Blend model during the model comparisons are shown in Table B-3. The values are chosen for these parameters and for the quasi-nonsingular ROE parameters in Section B-1-2 to obtain similar performance for all three models. This leads to the odd terms with n_c and r_c, where, for example, the equivalence between the HCW and the Blend model is used for the in-plane states[1].

Table B-3: Control parameters for Blend model during model comparisons.

Parameter	Description	Value
Q	State cost	$\texttt{diag}(25, 2500, \frac{1}{4}n_c^2\,10^4, n_c^2 r_c^2\,2500, 100, 100)$
Q_T	Terminal state cost	Q
R	Input cost	$\texttt{diag}(10^{-4}, 10^{-4}, 10^{-4})$
$x_{\max}$	Maximum state values	$\begin{bmatrix}0.1 & 10 & 0.02 & \frac{1}{100\,n_c\,r_c} & 1 & 1\end{bmatrix}^\mathsf{T}$
$x_{\min}$	Minimum state values	$-x_{\max}$
$u_{\max}$	Maximum input values	$\begin{bmatrix}0.1 & 0.1 & 0.1\end{bmatrix}^\mathsf{T}$
$u_{\min}$	Minimum input values	$-u_{\max}$
T	Prediction horizon	20

[1] See the end of Section A-3-3 for this proof.

B-2 System-Level Synthesis

For the SLS simulations, the Blend model is used with the parameters from Table B-4. This includes two noteworthy differences compared to Table B-3, as the prediction horizon is decreased for computational purposes and the terminal cost increased to account for the loss of long-term state costs.

Table B-4: Control parameters for Blend model in System-Level Synthesis results.

Parameter	Description	Value
Q	State cost	$\texttt{diag}(25, 2500, \frac{1}{4}n_c^2\, 10^4, n_c^2 r_c^2\, 2500, 100, 100)$
Q_T	Terminal state cost	$5Q$
R	Input cost	$\texttt{diag}(10^{-4}, 10^{-4}, 10^{-4})$
$x_{\max}$	Maximum state values	$\begin{bmatrix} 0.1 & 10 & 0.02 & \frac{1}{100\, n_c\, r_c} & 1 & 1 \end{bmatrix}^{\mathsf{T}}$
$x_{\min}$	Minimum state values	$-x_{\max}$
$u_{\max}$	Maximum input values	$\begin{bmatrix} 0.1 & 0.1 & 0.1 \end{bmatrix}^{\mathsf{T}}$
$u_{\min}$	Minimum input values	$-u_{\max}$
T	Prediction horizon	6

Appendix C

Simulation Setup

A vital part of the simulation is the integration setup that simulates the true, nonlinear dynamics of the satellite. This setup is covert in Appendix C-1, after which the selection for initial positions and reference states is discussed in Appendix C-2.

C-1 Integration Setup

The basis of the simulation is a mean orbital elements simulation. Because the initial and reference states are in a perfectly circular orbit, it is impossible to use Eq. (2-14) directly, as the derivatives for M and ω are not well defined for those states. Instead, the simulation is performed with the following state:

$$\mathbf{x}_{\text{sim}} = \begin{bmatrix} r & \theta & e\cos f & e\sin f & i & \Omega \end{bmatrix}^{\mathsf{T}},$$

with the following first derivative:

$$\dot{\mathbf{x}}_{\text{sim}} = f(\mathbf{x}_{\text{sim}}, t) = \begin{bmatrix} \sqrt{\frac{\mu(1+e\cos f)}{r}} e\sin f + \frac{re\sin f}{1+e\cos f}\dot{f}_{\mathrm{J}_2} \\ \sqrt{\frac{\mu(1+e\cos f)}{r^3}} - \frac{\sqrt{1-e^2}\sin(\theta)}{an(1+e\cos f)\tan i}u_n(t) + \dot{\omega}_{\mathrm{J}_2} + \dot{f}_{\mathrm{J}_2} \\ -n(1+e\cos f)^2\frac{e\sin f}{(1-e^2)^{\frac{3}{2}}} + 2\frac{\sqrt{1-e^2}}{an}u_t(t) - e\sin f\dot{f}_{\mathrm{J}_2} \\ \frac{na^2\sqrt{1-e^2}}{r^2}e\cos f + \frac{\sqrt{1-e^2}}{an}u_r(t) + \frac{\sqrt{1-e^2}e\sin f}{an(1+e\cos f)}u_t(t) + e\cos f\dot{f}_{\mathrm{J}_2} \\ \frac{\sqrt{1-e^2}\cos\theta}{an(1+e\cos f)}u_n(t) \\ \frac{\sqrt{1-e^2}\sin\theta}{an(1+e\cos f)\sin i}u_n(t) + \dot{\Omega}_{\mathrm{J}_2} \end{bmatrix},$$

which is constructed with help from Eqs. (A-6), (A-10), (A-12), (A-13), (A-19) and (2-14). The differential equations are integrated using the fourth-order Runge-Kutta (RK4) method, which follows the following update steps:

$$\begin{aligned} x_{\text{sim},k+1} &= x_{\text{sim},k} + \frac{\Delta t_{\text{sim}}}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\ t_{k+1} &= t_k + \Delta t_{\text{sim}}, \end{aligned}$$

where k_1, k_2, k_3 and k_4 are defined as:

$$\begin{aligned} k_1 &:= f(x_{\text{sim},k},\ t_k), \\ k_2 &:= f(x_{\text{sim},k} + \frac{\Delta t_{\text{sim}}}{2}k_1,\ t_k + \frac{\Delta t_{\text{sim}}}{2}), \\ k_3 &:= f(x_{\text{sim},k} + \frac{\Delta t_{\text{sim}}}{2}k_2,\ t_k + \frac{\Delta t_{\text{sim}}}{2}), \\ k_4 &:= f(x_{\text{sim},k} + \Delta t_{\text{sim}}\, k_3,\ t_k + \Delta t_{\text{sim}}). \end{aligned}$$

In these equations, Δt_{sim} denoted the integration time step.

C-2 Selecting Initial and Reference States

The scenarios all represent the situation where several satellites drop out of the constellation, after which the remaining satellites have to redistribute themselves evenly. This means most initial and reference states are zero, except those corresponding to the argument of latitude. Given the number of satellites after the dropout to be N_{sat}, the number of satellites that drop out is equal to:

$$N_{\text{drop}} = \left\lceil \frac{N_{\text{sat}}}{10} \right\rceil .$$

The number of satellites before the dropout is thus equal to:

$$N_{\text{before}} = N_{\text{drop}} + N_{\text{sat}},$$

such that the initial starting angles of these satellites (for simplicity, denoted for a single-plane scenario) are equal to:

$$\theta_{\text{before}} = \{0, \frac{2\pi}{N_{\text{before}}}, 2 \cdot \frac{2\pi}{N_{\text{before}}}, \dots, (N_{\text{before}} - 1) \cdot \frac{2\pi}{N_{\text{before}}}\}.$$

Of these N_{before} angles, N_{sat} angles are sampled without replacement to obtain the initial arguments of latitude at the start of the simulation.

The reference position for each satellite follows a similar pattern, where one starts with:

$$\theta_{\text{end}} = \{0, \frac{2\pi}{N_{\text{sat}}}, 2 \cdot \frac{2\pi}{N_{\text{sat}}}, \dots, (N_{\text{sat}} - 1) \cdot \frac{2\pi}{N_{\text{sat}}}\}.$$

However, to shorten the simulation time and divide the required workload among all satellites, the reference arguments of latitude are selected such that the mean initial error over all satellites is zero. That is, given a mean initial error $\bar{\theta}_{\text{err}}$:

$$\bar{\theta}_{\text{err}} = \frac{1}{N_{\text{sat}}} \sum_{i=1}^{N_{\text{sat}}} \theta^i_{\text{end}} - \theta^i_{\text{before}},$$

the reference states are given as:

$$\theta_{\text{ref}} = \{-\bar{\theta}_{\text{err}}, \frac{2\pi}{N_{\text{sat}}} - \bar{\theta}_{\text{err}}, 2 \cdot \frac{2\pi}{N_{\text{sat}}} - \bar{\theta}_{\text{err}}, \dots, (N_{\text{sat}} - 1) \cdot \frac{2\pi}{N_{\text{sat}}} - \bar{\theta}_{\text{err}}\}.$$

Appendix D

Estimating Uncertainties

D-1 Estimating Model Errors

Before the performance of the different SLS methods can be tested, the model inaccuracies must be determined. For both methods, this requires an estimation of:

$$\epsilon_A^j := \left\|A^j - \hat{A}^j\right\|_1, \quad \epsilon_{B_2}^j := \left\|B_2^j - \hat{B}_2^j\right\|_1, \qquad \forall j \in \mathbb{Z}_0^{n_x-1},$$

as for the simple approach it holds by definition of $\|(\cdot)\|_{\infty\to\infty}$ that:

$$\left\|A - \hat{A}\right\|_{\infty\to\infty} = \max\{\left\|A^0 - \hat{A}^0\right\|_1, \dots, \left\|A^{n_x-1} - \hat{A}^{n_x-1}\right\|_1\},$$

and similarly for $\left\|B_2 - \hat{B}_2\right\|_{\infty\to\infty}$. To find these estimations, the following steps are followed:

- Firstly, a set of states for the Blend model from Section 2-2-3 and corresponding orbital elements is generated that approximates the entirety of the feasible region with the state and input constraints. As converting Blend states to orbital elements is difficult, this is done by sampling 10^6 points in the orbital elements space, converting them to the blend states and then discarding any infeasible points.

- For all the feasible states, the next state is estimated both with the linear model from Section 2-2-3, as well as with the nonlinear model from Section 2-3-1. The simulation with the latter is considered the true dynamics, and it uses a smaller timestep than the sampling time of the controller.

- With the state from the model $\hat{x}_{t+1}$, the state from the nonlinear simulation x_{t+1} and the initial state x_t, it is possible to estimate the model uncertainty for A as:

  $$\left\|A^j - \hat{A}^j\right\|_1 \approx \frac{|x_{t+1}^j - \hat{x}_{t+1}^j|}{\|x_t\|_\infty}.$$

  By taking the largest estimated model uncertainty, the worst-case model uncertainty inside the feasible region is obtained.

- For the input matrix B_2, a similar approach is followed, but the effect of the state dynamics itself has to be cancelled out to fully account for the error in B_2. Therefore, given the input u_t, the state with this input x_{t+1}^{input} and the state without input x_{t+1}^{no}, the uncertainty can be approximated as:

  $$\left\|B_2^j - \hat{B}_2^j\right\|_1 \approx \frac{|x_{t+1}^{\text{input},j} - x_{t+1}^{\text{no},j} - \hat{B}_2^j u_t|}{\|u_t\|_\infty}.$$

Once again, the largest value over all tested inputs and states is taken as the estimated model uncertainty.

Following these steps for same setup as in Section 2-3-1, the following model uncertainties are obtained:

$$\begin{aligned}
\left\|A^0 - \hat{A}^0\right\|_1 &= 2.0 \cdot 10^{-3}, & \left\|A^1 - \hat{A}^1\right\|_1 &= 2.1 \cdot 10^{-3}, & \left\|A^2 - \hat{A}^2\right\|_1 &= 2.9 \cdot 10^{-5}, \\
\left\|A^3 - \hat{A}^3\right\|_1 &= 5.0 \cdot 10^{-5}, & \left\|A^4 - \hat{A}^4\right\|_1 &= 1.5 \cdot 10^{-3}, & \left\|A^5 - \hat{A}^5\right\|_1 &= 8.6 \cdot 10^{-4}, \\
\left\|B_2^0 - \hat{B}_2^0\right\|_1 &= 3.1 \cdot 10^{-4}, & \left\|B_2^1 - \hat{B}_2^1\right\|_1 &= 2.3 \cdot 10^{-3}, & \left\|B_2^2 - \hat{B}_2^2\right\|_1 &= 8.1 \cdot 10^{-5}, \\
\left\|B_2^3 - \hat{B}_2^3\right\|_1 &= 1.3 \cdot 10^{-4}, & \left\|B_2^4 - \hat{B}_2^4\right\|_1 &= 6.0 \cdot 10^{-4}, & \left\|B_2^5 - \hat{B}_2^5\right\|_1 &= 9.1 \cdot 10^{-4}, \\
\left\|A - \hat{A}\right\|_{\infty \to \infty} &= 2.1 \cdot 10^{-3}, & \left\|B_2 - \hat{B}_2\right\|_{\infty \to \infty} &= 2.3 \cdot 10^{-3}.
\end{aligned}$$

D-2 Estimating Disturbances

The robust SLS formulations can also deal with disturbances. To add these to the simulation, the following steps are followed:

- For each simulation step, a random disturbance $\delta_{x,k} \overset{i.i.d.}{\sim} \mathcal{U}(\delta_{\min}, \delta_{\max})$ is sampled. Here, $\delta_{x,k}$, $\delta_{\min}$ and $\delta_{\max}$ all lie in $\mathbb{R}^{n_x}$, with n_x being the state size. This changes the RK4 equations from Appendix C-1 as follows:

$$x_{\text{sim},k+1} = f(x_{\text{sim},k}, \delta_{x,k}) := x_{\text{sim},k} + \frac{\Delta t_{\text{sim}}}{6}(k_1 + 2k_2 + 2k_3 + k_4) + \Delta t_{\text{sim}} \delta_{x,k}.$$

- These disturbances are now added to the simulations states, which, as discussed in Appendix C-1 are different than the Blend states. The conversion from x_{sim} to x_{Blend} can be denoted as:

$$\begin{aligned}
x_{\text{Blend},k+1} &= g(x_{\text{sim},k+1}) \\
&= g(f(x_{\text{sim},k}, \delta_{x,k})),
\end{aligned}$$

  such that the disturbance on the blend states can be identified as:

$$\delta_{x,k}^{\text{Blend}} = g(f(x_{\text{sim},k}, \delta_{x,k})) - g(f(x_{\text{sim},k}, 0)).$$

- For the same states as when determining the model uncertainty in Appendix D-1, both $\delta_{\min}$ and $\delta_{\max}$ are applied to find the upper bound in the disturbance for each state j:

$$\sigma_\delta^j := \max\left(|\delta_{x,k}^{\text{Blend},j}| \ \ \forall x_{\text{sim}} \in \mathcal{X}_{\text{feas}}, \ \ \forall \delta_{x,k} \in \{\delta_{\min}, \delta_{\max}\}\right).$$

- For $\delta_{\max} = -\delta_{\min} = \left[10^{-4}, 10^{-4}, 10^{-5}, 10^{-5}, 10^{-4}, 10^{-4}\right]^\mathsf{T}$, the following upper bounds are obtained:

$$\begin{aligned}
&\sigma_\delta^0 = 0.0016, \quad \sigma_\delta^1 = 0.0017, \quad \sigma_\delta^2 = 0.00011, \quad \sigma_\delta^3 = 0.00011, \\
&\sigma_\delta^4 = 0.00093, \quad \sigma_\delta^5 = 0.00088, \quad \sigma_\delta = \max\left(\sigma_\delta^0, \sigma_\delta^1, \sigma_\delta^2, \sigma_\delta^3, \sigma_\delta^4, \sigma_\delta^5\right) = 0.0017.
\end{aligned}$$

Appendix E

Optimsation Problem Reformulation

This appendix contains two parts. In the first, Appendix E-1, the standard SLS problem is transformed into a basic quadratic problem, where the focus lies on transforming the problem from using a matrix decision variable to a vector. In the second, Appendix E-2, the robust lumped SLS formulation is also transformed into a basic quadratic program, where the nonlinear constraints are rewritten into equivalent linear constraints.

E-1 Transformed System-Level Synthesis Problem

The original SLS problem is presented in Eq. (4-1):

$$\begin{aligned}
\underset{\boldsymbol{\Phi}_x^0,\, \boldsymbol{\Phi}_u^0}{\text{minimize}} \quad & x_0^\top {\boldsymbol{\Phi}_x^0}^\top \mathcal{Q} \boldsymbol{\Phi}_x^0 x_0 + x_0^\top {\boldsymbol{\Phi}_u^0}^\top \mathcal{R} \boldsymbol{\Phi}_u^0 x_0 \\
\text{subject to} \quad & \begin{bmatrix} I - Z\mathcal{A} & -Z\mathcal{B}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_x^0 \\ \boldsymbol{\Phi}_u^0 \end{bmatrix} = I^0, \\
& x_{\min} \leq \boldsymbol{\Phi}_x^0 x_0 \leq x_{\max}, \\
& u_{\min} \leq \boldsymbol{\Phi}_u^0 x_0 \leq u_{\max}.
\end{aligned}$$

$\boldsymbol{\Phi}_x^0$ and $\boldsymbol{\Phi}_u^0$ are matrix variables, and therefore, this problem cannot be solved by standard solvers for quadratic programming.

Furthermore, note that Eq. (4-1) contains essentially two different matrix multiplications with $\boldsymbol{\Phi}_x$ and $\boldsymbol{\Phi}_u$, namely those with the initial state x_0 and those for the SLP. These two types are dealt with independently in Sections E-1-2 and E-1-3, after first denoting and refreshing the notations in Section E-1-1. The resulting transformed problem is then presented in Section E-1-4.

E-1-1 Notations

First, recall that the first block column contains the following matrices:

$$\boldsymbol{\Phi}_x^0 = \begin{bmatrix} \Phi_x^{0,0} \\ \Phi_x^{1,1} \\ \vdots \\ \Phi_x^{T-1,T-1} \\ \Phi_x^{T,T} \end{bmatrix}, \quad \boldsymbol{\Phi}_u^0 = \begin{bmatrix} \Phi_u^{0,0} \\ \Phi_u^{1,1} \\ \vdots \\ \Phi_u^{T-1,T-1} \\ \Phi_u^{T,T} \end{bmatrix}.$$

Then, when the different entries of, for example, $\Phi_x^{i,j}$ are labelled as follows, given that the total state size is n_x:

$$\Phi_x^{i,j} = \begin{bmatrix} \varphi_0^{x,i,j} & \varphi_{n_x}^{x,i,j} & \cdots & \varphi_{(n_x-2)n_x}^{x,i,j} & \varphi_{(n_x-1)n_x}^{x,i,j} \\ \varphi_1^{x,i,j} & \varphi_{n_x+1}^{x,i,j} & \cdots & \varphi_{(n_x-2)n_x+1}^{x,i,j} & \varphi_{(n_x-1)n_x+1}^{x,i,j} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \varphi_{n_x-2}^{x,i,j} & \varphi_{2n_x-2}^{x,i,j} & \cdots & \varphi_{(n_x-2)n_x-2}^{x,i,j} & \varphi_{n_x^2-2}^{x,i,j} \\ \varphi_{n_x-1}^{x,i,j} & \varphi_{2n_x-1}^{x,i,j} & \cdots & \varphi_{(n_x-2)n_x-1}^{x,i,j} & \varphi_{n_x^2-1}^{x,i,j} \end{bmatrix},$$

the vector form of which is denoted by $\varphi_x^{i,j}$ defined as:

$$\varphi_x^{i,j} := \begin{bmatrix} \varphi_0^{x,i,j} & \varphi_1^{x,i,j} & \cdots & \varphi_{n_x^2-2}^{x,i,j} & \varphi_{n_x^2-1}^{x,i,j} \end{bmatrix}^\mathsf{T}.$$

The vector $\boldsymbol{\varphi}_x^0$ (and similarly for $\boldsymbol{\varphi}_u^0$) contains all these $\varphi_x^{i,j}$:

$$\boldsymbol{\varphi}_x^0 = \begin{bmatrix} \varphi_x^{0,0\mathsf{T}} & \varphi_x^{1,1\mathsf{T}} & \cdots & \varphi_x^{T-1,T-1\mathsf{T}} & \varphi_x^{T,T\mathsf{T}} \end{bmatrix}^\mathsf{T}.$$

E-1-2 States and Inputs for System-Level Synthesis

States and inputs are represented in the SLS formulation (without disturbances) as follows:

$$\mathbf{x} = \mathbf{\Phi}_x^0 x_0 = \begin{bmatrix} \Phi_x^{0,0} \\ \Phi_x^{1,1} \\ \vdots \\ \Phi_x^{T-1,T-1} \\ \Phi_x^{T,T} \end{bmatrix} x_0, \qquad \mathbf{u} = \mathbf{\Phi}_u^0 x_0 = \begin{bmatrix} \Phi_u^{0,0} \\ \Phi_u^{1,1} \\ \vdots \\ \Phi_u^{T-1,T-1} \\ \Phi_u^{T,T} \end{bmatrix} x_0,$$

which comes back both in the objective function as well as in the constraints. The individual multiplications between $\Phi_x^{i,j}$ and x_0 can be rewritten as follows:

$$\begin{aligned} \Phi_x^{i,j} x_0 &= \begin{bmatrix} \varphi_0^{x,i,j} & \varphi_{n_x}^{x,i,j} & \cdots & \varphi_{(n_x-2)n_x}^{x,i,j} & \varphi_{(n_x-1)n_x}^{x,i,j} \\ \varphi_1^{x,i,j} & \varphi_{n_x+1}^{x,i,j} & \cdots & \varphi_{(n_x-2)n_x+1}^{x,i,j} & \varphi_{(n_x-1)n_x+1}^{x,i,j} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \varphi_{n_x-2}^{x,i,j} & \varphi_{2n_x-2}^{x,i,j} & \cdots & \varphi_{(n_x-2)n_x-2}^{x,i,j} & \varphi_{n_x^2-2}^{x,i,j} \\ \varphi_{n_x-1}^{x,i,j} & \varphi_{2n_x-1}^{x,i,j} & \cdots & \varphi_{(n_x-2)n_x-1}^{x,i,j} & \varphi_{n_x^2-1}^{x,i,j} \end{bmatrix} \begin{bmatrix} x_0^0 \\ x_0^1 \\ \vdots \\ x_0^{n_x-2} \\ x_0^{n_x-1} \end{bmatrix} \\ &= \begin{bmatrix} \sum_{n=0}^{n_x-1} \varphi_{n\cdot n_x}^{x,i,j} x_0^n \\ \sum_{n=0}^{n_x-1} \varphi_{n\cdot n_x+1}^{x,i,j} x_0^n \\ \vdots \\ \sum_{n=0}^{n_x-1} \varphi_{(n+1)n_x-2}^{x,i,j} x_0^n \\ \sum_{n=0}^{n_x-1} \varphi_{(n+1)n_x-1}^{x,i,j} x_0^n \end{bmatrix} \\ &= \underbrace{\left(x_0^\mathsf{T} \otimes I_{n_x}\right)}_{K_x} \varphi_x^{i,j}, \end{aligned}$$

where following the same steps, $\Phi_u^{i,j} x_0$ has the following equivalent form:

$$\Phi_u^{i,j} x_0 = \underbrace{(x_0^\top \otimes I_{n_u})}_{K_u} \varphi_u^{i,j}.$$

However, the original problem in Eq. (4-1) contains the terms $\boldsymbol{\Phi}_x^0 x_0$ and $\boldsymbol{\Phi}_u^0 x_0$, and thus this expression needs to be expanded. This is easily done for $\boldsymbol{\Phi}_x^0 x_0$:

$$\begin{aligned}
\boldsymbol{\Phi}_x^0 x_0 &= \begin{bmatrix} \Phi_x^{0,0} \\ \Phi_x^{1,1} \\ \vdots \\ \Phi_x^{T-1,T-1} \\ \Phi_x^{T,T} \end{bmatrix} x_0 \\
&= \begin{bmatrix} K_x & & & & \\ & K_x & & & \\ & & \ddots & & \\ & & & K_x & \\ & & & & K_x \end{bmatrix} \begin{bmatrix} \varphi_x^{0,0} \\ \varphi_x^{1,1} \\ \vdots \\ \varphi_x^{T-1,T-1} \\ \varphi_x^{T,T} \end{bmatrix} \\
&= \mathcal{K}_x \boldsymbol{\varphi}_x^0,
\end{aligned}$$

and where $\boldsymbol{\Phi}_u^0 x_0$ yields a similar equivalent form:

$$\begin{aligned}
\boldsymbol{\Phi}_u^0 x_0 &= \begin{bmatrix} \Phi_u^{0,0} \\ \Phi_u^{1,1} \\ \vdots \\ \Phi_u^{T-1,T-1} \\ \Phi_u^{T,T} \end{bmatrix} x_0 \\
&= \begin{bmatrix} K_u & & & & \\ & K_u & & & \\ & & \ddots & & \\ & & & K_u & \\ & & & & K_u \end{bmatrix} \begin{bmatrix} \varphi_u^{0,0} \\ \varphi_u^{1,1} \\ \vdots \\ \varphi_u^{T-1,T-1} \\ \varphi_u^{T,T} \end{bmatrix} \\
&= \mathcal{K}_u \boldsymbol{\varphi}_u^0.
\end{aligned}$$

E-1-3 Rewriting the System-Level Parameterisation

The SLP also has to be rewritten. This constraint is originally posed in the following form:

$$\begin{bmatrix} I - Z\mathcal{A} & -Z\mathcal{B}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_x^0 \\ \boldsymbol{\Phi}_u^0 \end{bmatrix} = I^0,$$

and essentially combines the following two constraints:

$$\begin{aligned}
\Phi_x^{0,0} &= I_{n_x}, \\
\Phi_x^{t+1,t+1} &= A\Phi_x^{t,t} + B_2 \Phi_u^{t,t} \;\; \forall t \in \mathbb{Z}_0^{T-1}.
\end{aligned}$$

The first constraint can easily be transformed, as it requires flattening the identity matrix into $I_{n_x}^{\text{flat}}$:

$$\Phi_x^{0,0} = I_{n_x} \Leftrightarrow \varphi_x^{0,0} = I_{n_x}^{\text{flat}}.$$

A small toy example quickly shows the pattern that occurs when transforming the second constraint to $\boldsymbol{\varphi}_x$ and $\boldsymbol{\varphi}_u$:

$$\Phi_x^{t+1,t+1} = A\Phi_x^{t,t} + B_2\Phi_u^{t,t},$$

$$\begin{bmatrix} \varphi_0^{x,t+1,t+1} & \varphi_2^{x,t+1,t+1} \\ \varphi_1^{x,t+1,t+1} & \varphi_3^{x,t+1,t+1} \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} \begin{bmatrix} \varphi_0^{x,t,t} & \varphi_2^{x,t,t} \\ \varphi_1^{x,t,t} & \varphi_3^{x,t,t} \end{bmatrix} + \begin{bmatrix} B_{00} & B_{01} & B_{02} \\ B_{10} & B_{11} & B_{12} \end{bmatrix} \begin{bmatrix} \varphi_0^{u,t,t} & \varphi_3^{u,t,t} \\ \varphi_1^{u,t,t} & \varphi_4^{u,t,t} \\ \varphi_2^{u,t,t} & \varphi_5^{u,t,t} \end{bmatrix},$$

$$\begin{bmatrix} \varphi_0^{x,t+1,t+1} \\ \varphi_1^{x,t+1,t+1} \\ \varphi_2^{x,t+1,t+1} \\ \varphi_3^{x,t+1,t+1} \end{bmatrix} = (I_2 \otimes A) \begin{bmatrix} \varphi_0^{x,t,t} \\ \varphi_1^{x,t,t} \\ \varphi_2^{x,t,t} \\ \varphi_3^{x,t,t} \end{bmatrix} + (I_2 \otimes B_2) \begin{bmatrix} \varphi_0^{u,t,t} \\ \varphi_1^{u,t,t} \\ \varphi_2^{u,t,t} \\ \varphi_3^{u,t,t} \\ \varphi_4^{u,t,t} \\ \varphi_5^{u,t,t} \end{bmatrix},$$

such that, in general, it holds that:

$$\varphi_x^{t+1,t+1} = (I_{n_x} \otimes A)\varphi_x^{t,t} + (I_{n_x} \otimes B_2)\varphi_u^{t,t}.$$

For simplicity of notation, this is rewritten as:

$$\varphi_x^{t+1,t+1} = A^{\varphi}\varphi_x^{t,t} + B_2^{\varphi}\varphi_u^{t,t},$$

with A^{φ} and B_2^{φ} defined as:

$$A^{\varphi} := I_{n_x} \otimes A, \quad B_2^{\varphi} := I_{n_x} \otimes B_2.$$

The second constraint can thus equivalently be posed as follows $\forall t \in \mathbb{Z}_0^{T-1}$:

$$\Phi_x^{t+1,t+1} = A\Phi_x^{t,t} + B_2\Phi_u^{t,t} \quad \Leftrightarrow \quad \varphi_x^{t+1,t+1} = A^{\varphi}\varphi_x^{t,t} + B_2^{\varphi}\varphi_u^{t,t}.$$

By defining $\mathcal{A}^{\varphi}$, $\mathcal{B}_2^{\varphi}$ and $\mathcal{I}^{\varphi}$ as follows:

$$\mathcal{A}^{\varphi} := \begin{bmatrix} A^{\varphi} & & \\ & \ddots & \\ & & A^{\varphi} \\ & & & 0 \end{bmatrix}, \quad \mathcal{B}_2^{\varphi} := \begin{bmatrix} B_2^{\varphi} & & \\ & \ddots & \\ & & B_2^{\varphi} \\ & & & 0 \end{bmatrix}, \quad \mathcal{I}^{\varphi} := \begin{bmatrix} I_{n_x}^{\text{flat}} \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

the two transformed constraints can be combined to obtain a similar SLP as before:

$$\begin{bmatrix} I - Z\mathcal{A}^{\varphi} & -Z\mathcal{B}_2^{\varphi} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi}_x^0 \\ \boldsymbol{\varphi}_u^0 \end{bmatrix} = \mathcal{I}^{\varphi}.$$

E-1-4 Transformed System-Level Synthesis Problem

Combining the results from Sections E-1-2 and E-1-3, the following problem is equivalent to the original SLS problem:

$$\begin{aligned}
\underset{\varphi_x^0, \varphi_u^0}{\text{minimize}} \quad & {\varphi_x^0}^\top \mathcal{Q}^\varphi \varphi_x^0 + {\varphi_u^0}^\top \mathcal{R}^\varphi \varphi_u^0 \\
\text{subject to} \quad & \begin{bmatrix} I - Z\mathcal{A}^\varphi & -Z\mathcal{B}_2^\varphi \end{bmatrix} \begin{bmatrix} \varphi_x^0 \\ \varphi_u^0 \end{bmatrix} = \mathcal{I}^\varphi, \\
& x_{\min} \leq \mathcal{K}_x \varphi_x^0 \leq x_{\max}, \\
& u_{\min} \leq \mathcal{K}_u \varphi_u^0 \leq u_{\max},
\end{aligned}$$

where $\mathcal{Q}^\varphi$ and $\mathcal{R}^\varphi$ are defined as:

$$\mathcal{Q}^\varphi := \mathcal{K}_x^\top \mathcal{Q} \mathcal{K}_x, \quad \mathcal{R}^\varphi := \mathcal{K}_u^\top \mathcal{R} \mathcal{K}_u.$$

E-2 Robust System-Level Synthesis Problem

To work around this problem, first, a possible reformulation of infinity and 1-norms is provided in Section E-2-1. This is followed by the simplification of the constraints for the dynamical constraints (Section E-2-2), the lumped uncertainty bounds (Section E-2-3) and the constraint tightening (Section E-2-4).

E-2-1 Simplification of Norms

Firstly, recall the equation for the upper bound on the lumped uncertainty from Eq. (3-21):

$$\begin{aligned}
|\eta_t^j| \leq \epsilon_A^j & \left(\left\| \tilde{\Phi}_x^{t,t} x_0 \right\|_\infty + \sum_{i=1}^{t} \left\| \tilde{\Phi}_x^{t,t-i} \right\|_{\infty\to\infty} \right) \\
& + \epsilon_{B_2}^j \left(\left\| \tilde{\Phi}_u^{t,t} x_0 \right\|_\infty + \sum_{i=1}^{t} \left\| \tilde{\Phi}_u^{t,t-i} \right\|_{\infty\to\infty} \right) + \sigma_\delta^j \leq \sigma_t^j, \quad \forall j \in \mathbb{Z}_0^{n_x-1}, \quad \forall t \in \mathbb{Z}_1^{T-1}.
\end{aligned}$$

The infinity norms make it impossible to model this exact problem in, for example, `OSQP`, and slow solvers such as `Gurobi` down. However, it is possible to convert this equation to a standard quadratic problem using extra variables.

Secondly, recall that the infinity-norm and one-norm are defined as follows for a vector $x \in \mathbb{R}^N$ and matrix $A \in \mathbb{R}^{N\times M}$:

$$x := \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}, \quad A := \begin{bmatrix} A^1 \\ \vdots \\ A^N \end{bmatrix} = \begin{bmatrix} A_1^1 & \dots & A_M^1 \\ & \vdots & \\ A_1^N & \dots & A_M^N \end{bmatrix},$$

$$\|x\|_\infty := \max(|x_1|, \dots, |x_N|), \quad \|x\|_1 := \sum_{i=1}^{N} |x_i|,$$

$$\|A\|_{\infty\to\infty} := \left\| \begin{bmatrix} \|A^1\|_1 \\ \vdots \\ \|A^N\|_1 \end{bmatrix} \right\|_\infty = \max\left(\left\|A^1\right\|_1, \dots, \left\|A^N\right\|_1 \right).$$

For both the one-norm and the infinity-norm, it is required to find the absolute values of the vector or matrix. By splitting a variable x_t up into a positive and negative part, it is possible to find the absolute value in a quadratic (or linear) program:

$$\begin{aligned} |x_i| = \underset{x_i^+, x_i^-}{\text{minimize}} \quad & x_i^+ + x_i^- \\ \text{subject to} \quad & x_t = x_t^+ - x_t^-, \\ & x_i^+ \geq 0, \;\; x_i^- \geq 0. \end{aligned}$$

The extensions to the one-norm and infinity-norm can then easily be made:

$$\begin{aligned} \|x\|_1 = \underset{x^+, x^-}{\text{minimize}} \quad & \sum_{i=1}^{N} x_i^+ + x_i^- \\ \text{subject to} \quad & x = x^+ - x^-, \\ & x^+ \geq 0, \;\; x^- \geq 0. \end{aligned} \qquad \begin{aligned} \|x\|_\infty = \underset{x^+, x^-, x_{\max}}{\text{minimize}} \quad & x_{\max} \\ \text{subject to} \quad & x_i^+ + x_i^- \leq x_{\max} \;\; \forall i, \\ & x = x^+ - x^-, \\ & x^+ \geq 0, \;\; x^- \geq 0. \end{aligned}$$

E-2-2 Dynamics Constraints

The SLP from Eq. (3-16) is once more denoted below for convenience:

$$\begin{bmatrix} I - Z\hat{\mathcal{A}} & -Z\hat{\mathcal{B}}_2 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{\Phi}}_x \\ \tilde{\mathbf{\Phi}}_u \end{bmatrix} = \mathbf{\Sigma},$$

with $\mathbf{\Sigma}$ equal to:

$$\mathbf{\Sigma} = \begin{bmatrix} I & & & \\ & \Sigma_0 & & \\ & & \ddots & \\ & & & \Sigma_{T-1} \end{bmatrix}, \qquad \Sigma_t = \begin{bmatrix} \sigma_t^0 & & \\ & \ddots & \\ & & \sigma_t^{n_x - 1} \end{bmatrix}.$$

Although this constraint contains no norms, it is possible to simplify it slightly nonetheless. Firstly, the diagonal elements of $\tilde{\mathbf{\Phi}}_x$ are equal to:

$$\tilde{\Phi}_x^{0,0} = I, \quad \tilde{\Phi}_x^{t,0} = \Sigma_{t-1}, \;\; \forall t \in \mathbb{Z}_1^T.$$

Therefore, the dynamical constraint for $\tilde{\mathbf{\Phi}}_x^{1,1}$ can be rewritten as:

$$\tilde{\Phi}_x^{1,1} - \hat{A}\tilde{\Phi}_x^{0,0} - \hat{B}_2\tilde{\Phi}_u^{0,0} = 0 \Leftrightarrow \tilde{\Phi}_x^{1,1} - \hat{B}_2\tilde{\Phi}_u^{0,0} = A.$$

The same approach can be followed for the other diagonal block matrices of $\tilde{\mathbf{\Phi}}_x$, where instead of an identity matrix, the corresponding matrix Σ_{t-1} is used:

$$\tilde{\Phi}_x^{t+1,1} - \hat{B}_2\tilde{\Phi}_u^{t,0} = \hat{A}\Sigma_{t-1}, \;\; \forall t \in \mathbb{Z}_1^T.$$

For all other matrices, the equations remain the same:

$$\tilde{\Phi}_x^{i+1,j+1} - \hat{A}\tilde{\mathbf{\Phi}}_x^{i,j} - \hat{B}_2\tilde{\Phi}_u^{i,j} = 0, \;\; \forall i \in \mathbb{Z}_1^T, \;\; \forall j \in \mathbb{Z}_1^i.$$

The transformed forms of these equations, following the same steps as in Appendix E-1, are then as follows:

$$
\begin{aligned}
\tilde{\varphi}_x^{1,1} - \hat{B}_2^{\varphi}\tilde{\varphi}_u^{0,0} &= A^{\varphi} I_{nx}^{\text{flat}}, \\
\tilde{\varphi}_x^{t+1,1} - \hat{B}_2^{\varphi}\tilde{\varphi}_u^{t,0} &= A^{\varphi}\Sigma_{t-1}^{\text{flat}}, \quad \forall t \in \mathbb{Z}_1^T, \\
\tilde{\varphi}_x^{i+1,j+1} - \hat{A}^{\varphi}\tilde{\varphi}_x^{i,j} - \hat{B}_2^{\varphi}\tilde{\varphi}_u^{i,j} &= 0, \quad \forall i \in \mathbb{Z}_1^T, \quad \forall j \in \mathbb{Z}_1^i,
\end{aligned}
$$

which can equivalently be presented in the common SLP form:

$$
\begin{bmatrix} I - Z\mathcal{A}^{\varphi} & -\mathcal{B}_2^{\varphi} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\varphi}}_x^t \\ \tilde{\boldsymbol{\varphi}}_u^t \end{bmatrix} = \mathcal{I}_t^{\varphi}, \quad \forall t \in \mathbb{Z}_0^T,
$$

with $\tilde{\boldsymbol{\varphi}}_x^t$, $\tilde{\boldsymbol{\varphi}}_u^t$ and $\mathcal{I}_t^{\varphi}$ defined as:

$$
\tilde{\boldsymbol{\varphi}}_x^t := \begin{bmatrix} \tilde{\varphi}_x^{t+1,1} \\ \vdots \\ \tilde{\varphi}_x^{T,T-t} \end{bmatrix}, \quad \tilde{\boldsymbol{\varphi}}_u^t := \begin{bmatrix} \tilde{\varphi}_u^{t,0} \\ \vdots \\ \tilde{\varphi}_u^{T-1,T-t-1} \end{bmatrix},
$$

$$
\mathcal{I}_0^{\varphi} := A^{\varphi} \begin{bmatrix} I_{n_x}^{\text{flat}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathcal{I}_t^{\varphi} := A^{\varphi} \begin{bmatrix} \Sigma_{t-1}^{\text{flat}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ \forall t \in \mathbb{Z}_1^T.
$$

Note that the indices for the entries of $\tilde{\boldsymbol{\varphi}}_x^t$ are one higher than before, as the values for $\tilde{\Phi}_x^{t,0}$ are known beforehand and have been moved into $\mathcal{I}_t^{\varphi}$. As the input at the terminal time step has no effect on the states, the last term from $\tilde{\boldsymbol{\varphi}}_u^t$ can be removed such that $\tilde{\boldsymbol{\varphi}}_x^t$ and $\tilde{\boldsymbol{\varphi}}_u^t$ correspond to the same number of time steps. However, because $\tilde{\boldsymbol{\varphi}}_x^t$ is shifted, the matrix Z is dropped in the SLP for $\tilde{\boldsymbol{\varphi}}_u^t$.

The sparse form of this SLP is easily found with the same procedure as in Section 4-1-3:

$$
\begin{bmatrix} (I - Z\mathcal{A}^{\varphi})[i_{\text{sp},t}^x, i_{\text{sp},t}^x] & -\mathcal{B}_2^{\varphi}[i_{\text{sp},t}^x, i_{\text{sp},t}^u] \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\varphi}}_x^t \\ \tilde{\boldsymbol{\varphi}}_u^t \end{bmatrix} = \mathcal{I}_t^{\varphi}[i_{\text{sp},t}^x], \quad \forall t \in \mathbb{Z}_0^T,
$$

with $i_{\text{sp},t}^x$ and $i_{\text{sp},t}^u$ corresponding to the non-zero elements of $\tilde{\boldsymbol{\varphi}}_x^t$ and $\tilde{\boldsymbol{\varphi}}_u^t$.

E-2-3 Lumped Uncertainty Bound

The lumped uncertainty bound are given in Eqs. (3-20) and (3-21):

$$
\begin{aligned}
\left|\eta_0^j\right| &\le \epsilon_A^j \left\|\tilde{\Phi}_x^{0,0} x_0\right\|_\infty + \epsilon_{B_2}^j \left\|\tilde{\Phi}_u^{0,0} x_0\right\|_\infty + \sigma_\delta^j \le \sigma_0^j, \quad \forall j \in \mathbb{Z}_0^{n_x - 1}, \\
|\eta_t^j| &\le \epsilon_A^j \left(\left\|\tilde{\Phi}_x^{t,t} x_0\right\|_\infty + \sum_{i=1}^{t} \left\|\tilde{\Phi}_x^{t,t-i}\right\|_{\infty\to\infty} \right) \\
&\quad + \epsilon_{B_2}^j \left(\left\|\tilde{\Phi}_u^{t,t} x_0\right\|_\infty + \sum_{i=1}^{t} \left\|\tilde{\Phi}_u^{t,t-i}\right\|_{\infty\to\infty} \right) + \sigma_\delta^j \le \sigma_t^j, \quad \forall j \in \mathbb{Z}_0^{n_x - 1}, \quad \forall t \in \mathbb{Z}_1^{T-1},
\end{aligned}
$$

and requires the simplification of the infinity-norms before it can be used in a standard quadratic problem. Firstly, recall with help from Appendix E-1 that $\tilde{\Phi}_x^{t,t}x_0$ and $\tilde{\Phi}_u^{t,t}x_0$ are equal to:

$$x_t = \tilde{\Phi}_x^{t,t}x_0 = K_x\tilde{\varphi}_x^{t,t}, \quad u_t = \tilde{\Phi}_u^{t,t}x_0 = K_u\tilde{\varphi}_u^{t,t}.$$

Then, the simplified version of these infinity-norms can be found using the theory from Section E-2-1:

$$\begin{aligned} \left\|\tilde{\Phi}_x^{t,t}x_0\right\|_\infty = \underset{x_t^+, x_t^-, x_t^{\max}}{\text{minimize}} \quad & x_t^{\max} \\ \text{subject to} \quad & x_{t,i}^+ + x_{t,i}^- \leq x_t^{\max}, \quad \forall i \in \mathbb{Z}_0^{n_x-1}, \\ & K_x\tilde{\varphi}^{t,t} = x_t^+ - x_t^-, \\ & x_t^+ \geq 0, \quad x_t^- \geq 0. \end{aligned} \tag{E-1}$$

where the result for $\tilde{\Phi}_u^{t,t}x_0$ is analogous to the result presented above.

With $\tilde{\Phi}_x^{t,t-i,j}$ denoting the j'th row of $\tilde{\Phi}_x^{t,t-i}$ and e_j the j'th standard basis, its one-norm is equal to:

$$\left\|\tilde{\Phi}_x^{t,t-i,j}\right\|_1 = \underbrace{\mathbf{1}_{n_x}^\top \otimes e_j^\top}_{N_x^j}\left|\tilde{\varphi}_x^{t,t-i}\right|. \tag{E-2}$$

This allows for the computation of $\left\|\tilde{\Phi}_x^{t,t-i}\right\|_{\infty\to\infty}$, where simply the maximum one-norm has to be selected:

$$\begin{aligned} \left\|\tilde{\Phi}_x^{t,t-i}\right\|_{\infty\to\infty} = \underset{\substack{\tilde{\varphi}_x^{t,t-i,+}, \tilde{\varphi}_x^{t,t-i,-}, \\ \tilde{\varphi}_{x,\max}^{t,t-i}}}{\text{minimize}} \quad & \tilde{\varphi}_{x,\max}^{t,t-i} \\ \text{subject to} \quad & N_x^j(\tilde{\varphi}_x^{t,t-i,+} + \tilde{\varphi}_x^{t,t-i,-}) \leq \tilde{\varphi}_{x,\max}^{t,t-i}, \quad \forall j \in \mathbb{Z}_0^{n_x-1}, \\ & \tilde{\varphi}_x^{t,t-i} = \tilde{\varphi}_x^{t,t-i,+} - \tilde{\varphi}_x^{t,t-i,-}, \\ & \tilde{\varphi}_x^{t,t-i,+} \geq 0, \quad \tilde{\varphi}_x^{t,t-i,-} \geq 0. \end{aligned} \tag{E-3}$$

A similar formulation can be derived for the infinity-norm of $\tilde{\Phi}_u^{t,t-i}$. This is in contrast to $\tilde{\Phi}_x^{t,0}$, as these matrices were removed from the dynamical constraint in Section E-2-2. Luckily, it is fairly straightforward to see that:

$$\begin{aligned} \left\|\tilde{\Phi}_x^{t,0}\right\|_{\infty\to\infty} = \|\Sigma_{t-1}\|_{\infty\to\infty} \underset{\sigma_{t-1}^{\max}}{\text{minimize}} \quad & \sigma_{t-1}^{\max} \\ \text{subject to} \quad & \sigma_{t-1}^i \leq \sigma_{t-1}^{\max}, \quad \forall i \in \mathbb{Z}_0^{n_x-1} \end{aligned} \tag{E-4}$$

as σ_{t-1}^i is guaranteed to be larger or equal to zero and as Σ_{t-1} is a diagonal matrix, making its diagonal elements equal to the one-norm of its row. Combining all these results, it is possible to find the equivalent formulation of the lumped uncertainty bound:

$$
\begin{aligned}
\epsilon_A^j x_0^{\max} + e_{B_2}^j u_0^{\max} + \sigma_\delta^j \leq \sigma_0^j, \quad \forall j \in \mathbb{Z}_0^{n_x-1}, \\
\epsilon_A^j \left(x_1^{\max} + \sigma_0^{\max}\right) + e_{B_2}^j \left(u_1^{\max} + \tilde{\varphi}_{u,\max}^{1,0}\right) + \sigma_\delta^j \leq \sigma_1^j, \quad \forall j \in \mathbb{Z}_0^{n_x-1}, \\
\epsilon_A^j \left(x_t^{\max} + \sum_{i=1}^{t-1} \tilde{\varphi}_{x,\max}^{t,t-i} + \sigma_{t-1}^{\max}\right) \\
+ \epsilon_{B_2}^j \left(u_t^{\max} + \sum_{i=1}^{t} \tilde{\varphi}_{u,\max}^{t,t-i}\right) + \sigma_\delta^j \leq \sigma_t^j, \quad \forall j \in \mathbb{Z}_0^{n_x-1}, \quad \forall t \in \mathbb{Z}_2^{T-1},
\end{aligned}
\tag{E-5}
$$

where σ_t^j must be minimised to find the correct solution:

$$
\begin{aligned}
\text{minimize} \quad & \sigma_t^j \\
\text{subject to} \quad & \text{Constraints for } x_t^{\max} \text{ and } u_t^{\max} \text{ following Eq. (E-1)}, \\
& \text{Constraints for } \tilde{\varphi}_{x,\max}^{t,t-i} \text{ and } \tilde{\varphi}_{u,\max}^{t,t-i} \text{ following Eq. (E-3)}, \\
& \text{Constraints for } \sigma_{t-1}^{\max} \text{ following Eq. (E-4)}, \\
& \text{Lumped uncertainty bound from Eq. (E-5)}.
\end{aligned}
$$

E-2-4 Constraint Tightening

The ILSLS tightened constraints are shown in Eq. (3-22):

$$
\begin{aligned}
H_{\mathcal{X}_t}^j \tilde{\Phi}_x^{t,t} x_0 + \sum_{i=1}^{t-1} \left\| H_{\mathcal{X}_t}^j \tilde{\Phi}_x^{t,t-i} \right\|_1 + \left\| H_{\mathcal{X}_t}^j \Sigma_{t-1} \right\|_1 \leq h_{\mathcal{X}_t}^j, \ \forall j \in \mathbb{Z}_0^{n\mathcal{X}_t - 1}, \ \forall t \in \mathbb{Z}_1^{T-1}, \\
H_{\mathcal{X}_T}^j \tilde{\Phi}_x^{T,T} x_0 + \sum_{i=1}^{T-1} \left\| H_{\mathcal{X}_T}^j \tilde{\Phi}_x^{T,T-i} \right\|_1 + \left\| H_{\mathcal{X}_T}^j \Sigma_{T-1} \right\|_1 \leq h_{\mathcal{X}_T}^j, \ \forall j \in \mathbb{Z}_0^{n\mathcal{X}_T}, \\
H_{\mathcal{U}_t}^j \tilde{\Phi}_u^{t,t} x_0 + \sum_{i=1}^{t} \left\| H_{\mathcal{U}_t}^j \tilde{\Phi}_u^{t,t-i} \right\|_1 \leq h_{\mathcal{U}_t}^j, \ \forall j \in \mathbb{Z}_0^{n\mathcal{U}_t - 1}, \ \forall t \in \mathbb{Z}_0^{T-1},
\end{aligned}
$$

such that it holds that:

$$
H_{\mathcal{X}_t} x_t \leq h_{\mathcal{X}_t}, \quad H_{\mathcal{X}_T} x_T \leq h_{\mathcal{X}_T} \quad H_{\mathcal{U}_t} u_t \leq h_{\mathcal{U}_t}.
$$

The state and input constraints limit these such that $x_{\min} \leq x_t \leq x_{\max}$ and $u_{\min} \leq u_t \leq u_{\max}$. This means that $H_{\mathcal{X}_t}$, $H_{\mathcal{X}_T}$ and $H_{\mathcal{U}_t}$, along with $h_{\mathcal{X}_t}$, $h_{\mathcal{X}_T}$ and $h_{\mathcal{U}_t}$ are set up as:

$$
\begin{bmatrix} -I_{n_x} \\ I_{n_x} \end{bmatrix} x_t \leq \begin{bmatrix} -x_{\min} \\ x_{\max} \end{bmatrix}, \quad \begin{bmatrix} -I_{n_x} \\ I_{n_x} \end{bmatrix} x_T \leq \begin{bmatrix} -x_{\min} \\ x_{\max} \end{bmatrix}, \quad \begin{bmatrix} -I_{n_u} \\ I_{n_u} \end{bmatrix} u_t \leq \begin{bmatrix} -u_{\min} \\ u_{\max} \end{bmatrix}.
$$

$H_{\mathcal{X}_t}^j$ (and similar for $H_{\mathcal{X}_T}^j$ and $H_{\mathcal{U}_t}^j$) is closely connected to the standard basis e_j:

$$
H_{\mathcal{X}_t}^j = \begin{cases} -e_{m_j^x}, & \text{if } j < n_x, \\ e_{m_j^x} & \text{if } j \geq n_x, \end{cases}
$$

where m_j^x uses the modulo operator and is defined as:

$$m_j^x := j \bmod n_x.$$

Using this, the one norms simplify:

$$\begin{aligned}
&\left\| H^j_{\mathcal{X}_t} \tilde{\Phi}_x^{t,t-i} \right\|_1 = \left\| e_{m_j^x} \tilde{\Phi}_x^{t,t-i} \right\|_1 \vee \left\| H^j_{\mathcal{X}_t} \tilde{\Phi}_x^{t,t-i} \right\|_1 = \left\| -e_{m_j^x} \tilde{\Phi}_x^{t,t-i} \right\|_1 \\
&\left\| H^j_{\mathcal{X}_t} \tilde{\Phi}_x^{t,t-i} \right\|_1 = \left\| \tilde{\Phi}_x^{t,t-i,m_j^x} \right\|_1 \vee \left\| H^j_{\mathcal{X}_t} \tilde{\Phi}_x^{t,t-i} \right\|_1 = \left\| -\tilde{\Phi}_x^{t,t-i,m_j^x} \right\|_1 \\
&\left\| H^j_{\mathcal{X}_t} \tilde{\Phi}_x^{t,t-i} \right\|_1 = \left\| \tilde{\Phi}_x^{t,t-i,m_j^x} \right\|_1 ,
\end{aligned}$$

where $\left\| \tilde{\Phi}_x^{t,t-i,m_j^x} \right\|_1$ was already simplified in Eq. (E-2). Thus, for the states this yields the following constraints for each row j of $H_{\mathcal{X}_t}$:

$$\begin{aligned}
\text{minimize} \quad & N_x^{m_j^x} \sum_{i=1}^{t-1} (\tilde{\varphi}_x^{t,t-i,+} + \tilde{\varphi}_x^{t,t-i,-}) \\
\text{subject to} \quad & H^j_{\mathcal{X}_t} K_x \tilde{\varphi}_x^{t,t} + \sum_{i=1}^{t-1} N_x^{m_j^x} (\tilde{\varphi}_x^{t,t-i,+} + \tilde{\varphi}_x^{t,t-i,-}) + \sigma_{t-1}^{m_j^x} \leq h^j_{\mathcal{X}_t}, \ \ \forall t \in \mathbb{Z}_1^{T-1}, \\
& \tilde{\varphi}_x^{t,t-i} = \tilde{\varphi}_x^{t,t-i,+} - \tilde{\varphi}_x^{t,t-i,-}, \ \ \forall i \in \mathbb{Z}_1^{t-1}, \ \ \forall t \in \mathbb{Z}_1^{T-1}, \\
& \tilde{\varphi}_x^{t,t-i,+} \geq 0, \ \ \tilde{\varphi}_x^{t,t-i,-} \geq 0, \ \ \forall i \in \mathbb{Z}_1^{t-1}, \ \ \forall t \in \mathbb{Z}_1^{T-1},
\end{aligned}$$

with the results for $H_{\mathcal{X}_T}$ and $H_{\mathcal{U}_t}$ being elementary variants of these equations. The latter differs the most, as $\tilde{\Phi}_u^{t,0}$ cannot be replaced by Σ_{t-1}:

$$\begin{aligned}
\text{minimize} \quad & N_u^{m_j^u} \sum_{i=1}^{t} (\tilde{\varphi}_u^{t,t-i,+} + \tilde{\varphi}_u^{t,t-i,-}) \\
\text{subject to} \quad & H^j_{\mathcal{U}_t} K_u \tilde{\varphi}_u^{t,t} + \sum_{i=1}^{t} N_u^{m_j^u} (\tilde{\varphi}_u^{t,t-i,+} + \tilde{\varphi}_u^{t,t-i,-}) \leq h^j_{\mathcal{U}_t}, \ \ \forall t \in \mathbb{Z}_0^{T-1}, \\
& \tilde{\varphi}_u^{t,t-i} = \tilde{\varphi}_u^{t,t-i,+} - \tilde{\varphi}_u^{t,t-i,-}, \ \ \forall i \in \mathbb{Z}_1^{t}, \ \ \forall t \in \mathbb{Z}_0^{T-1}, \\
& \tilde{\varphi}_u^{t,t-i,+} \geq 0, \ \ \tilde{\varphi}_u^{t,t-i,-} \geq 0, \ \ \forall i \in \mathbb{Z}_1^{t}, \ \ \forall t \in \mathbb{Z}_0^{T-1}.
\end{aligned}$$

Appendix F

Extensive Simulation Results

This chapter extends the results from simulations in Sections 2-3, 3-4 and 5-3. All controller states and inputs are fully provided for each simulation.

F-1 Model Comparison With Planar Simulation

F-1-1 Hill-Clohessy-Wiltshire Model

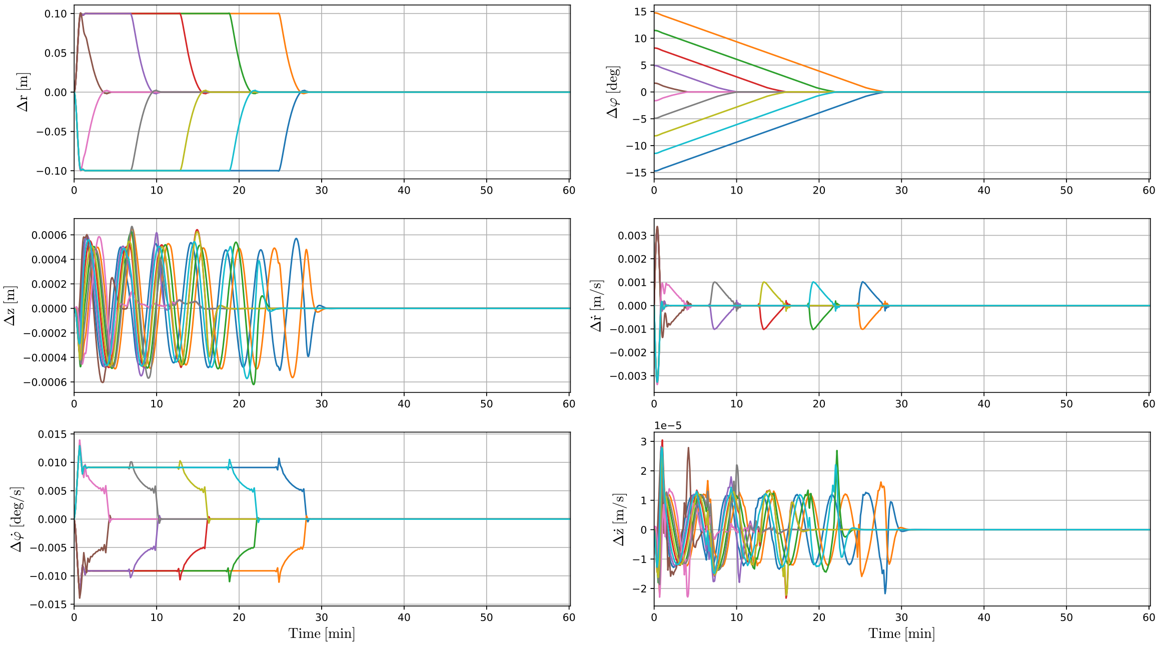


Figure F-1: States for Hill-Clohessy-Wiltshire model in planar scenario with J_2 perturbations.

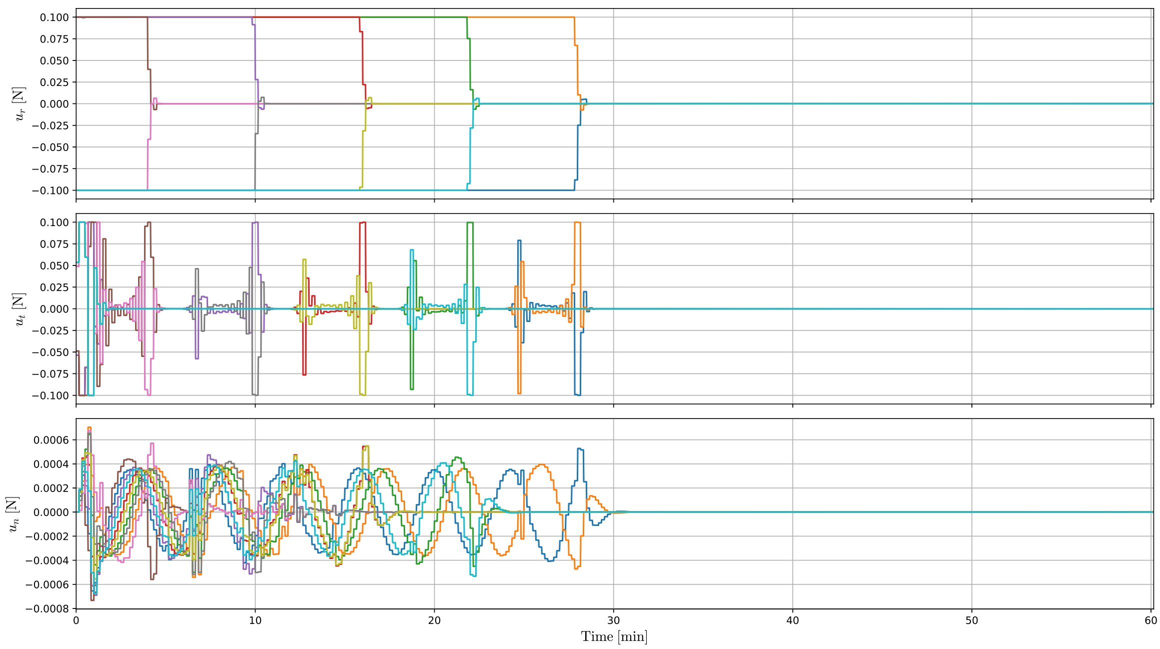


Figure F-2: Inputs for Hill-Clohessy-Wiltshire model in planar scenario with J_2 perturbations.

F-1-2 Relative Orbital Elements Model

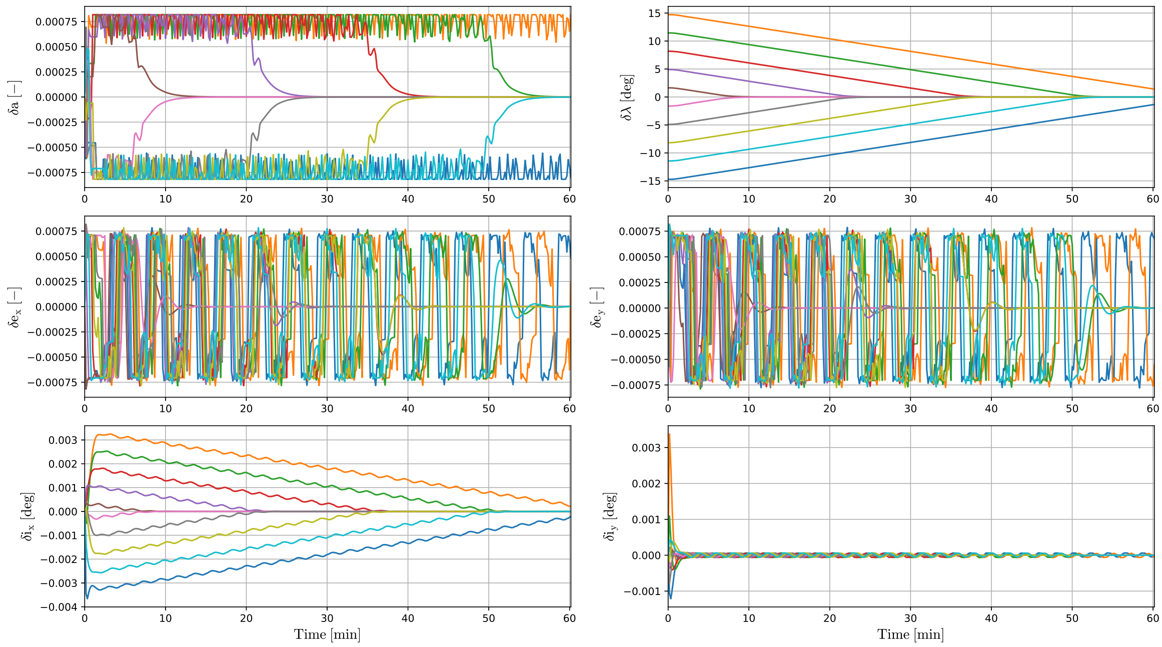


Figure F-3: States for Relative Orbital Elements model in planar scenario with J_2 perturbations.

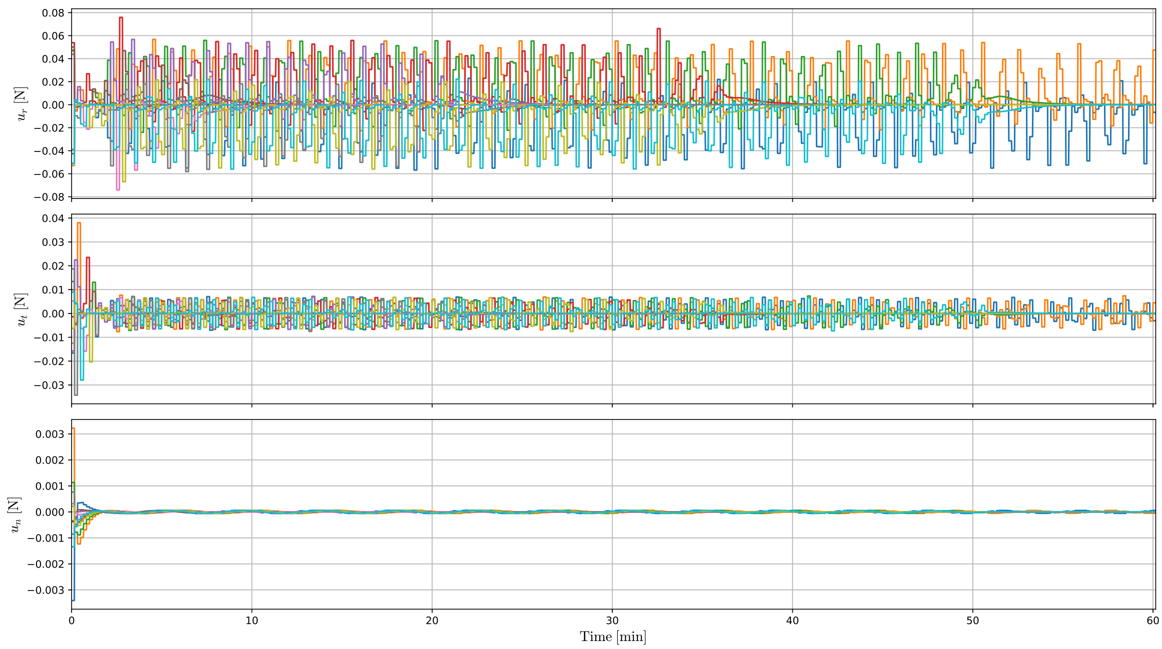


Figure F-4: Inputs for Relative Orbital Elements model in planar scenario with J_2 perturbations.

F-1-3 Blend Model

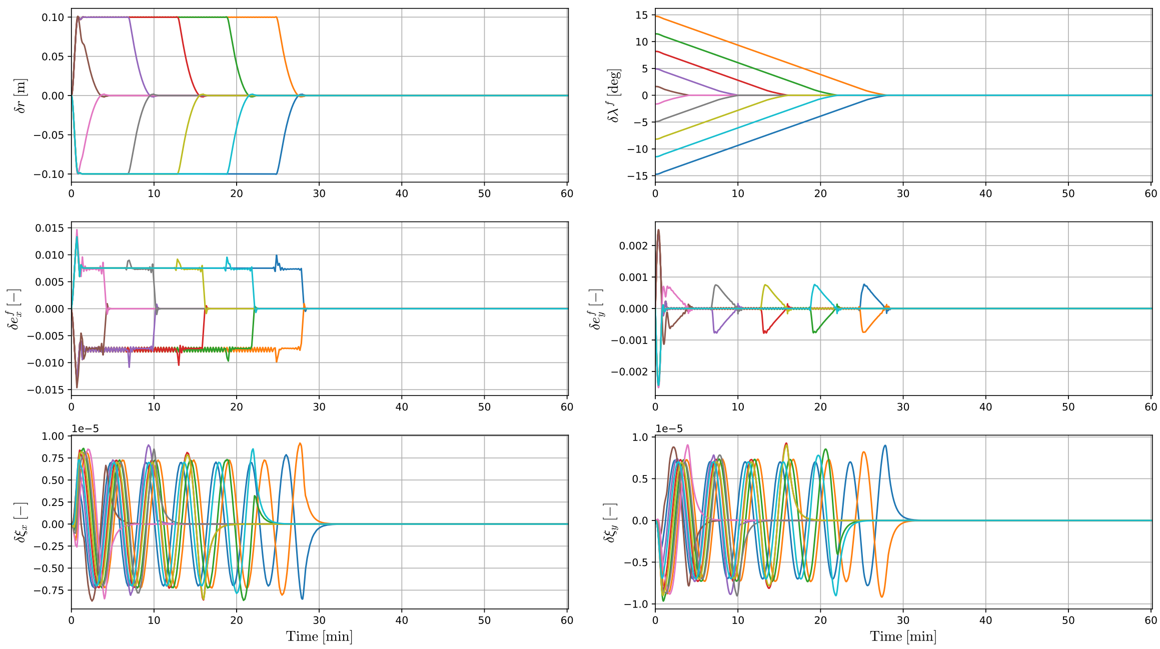


Figure F-5: States for Blend model in planar scenario with J_2 perturbations.

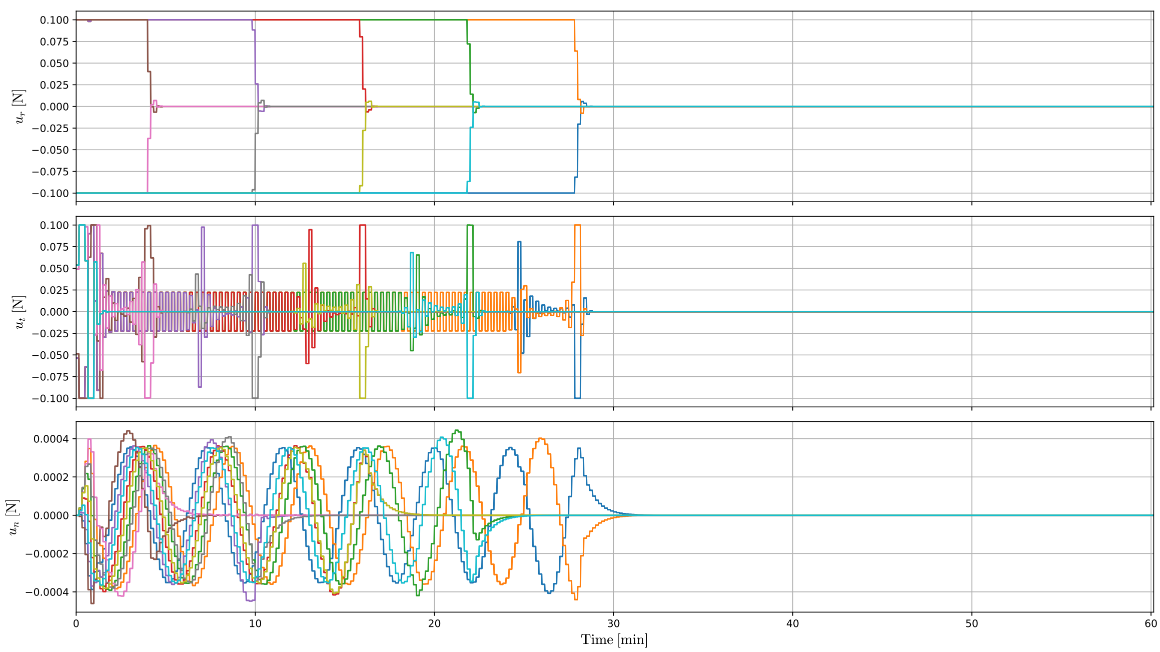


Figure F-6: Inputs for Blend model in planar scenario with J_2 perturbations.

F-2 Model Comparison With Small Inter-Planar Simulation

F-2-1 Hill-Clohessy-Wiltshire Model

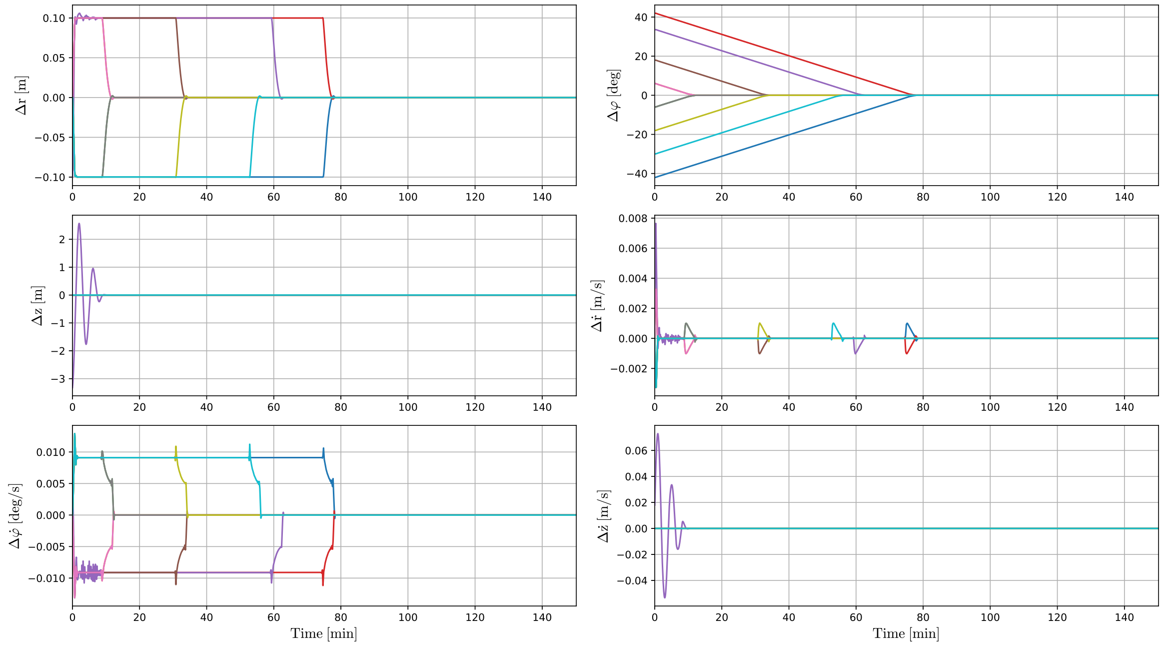


Figure F-7: States for Hill-Clohessy-Wiltshire model in small inter-planar scenario.

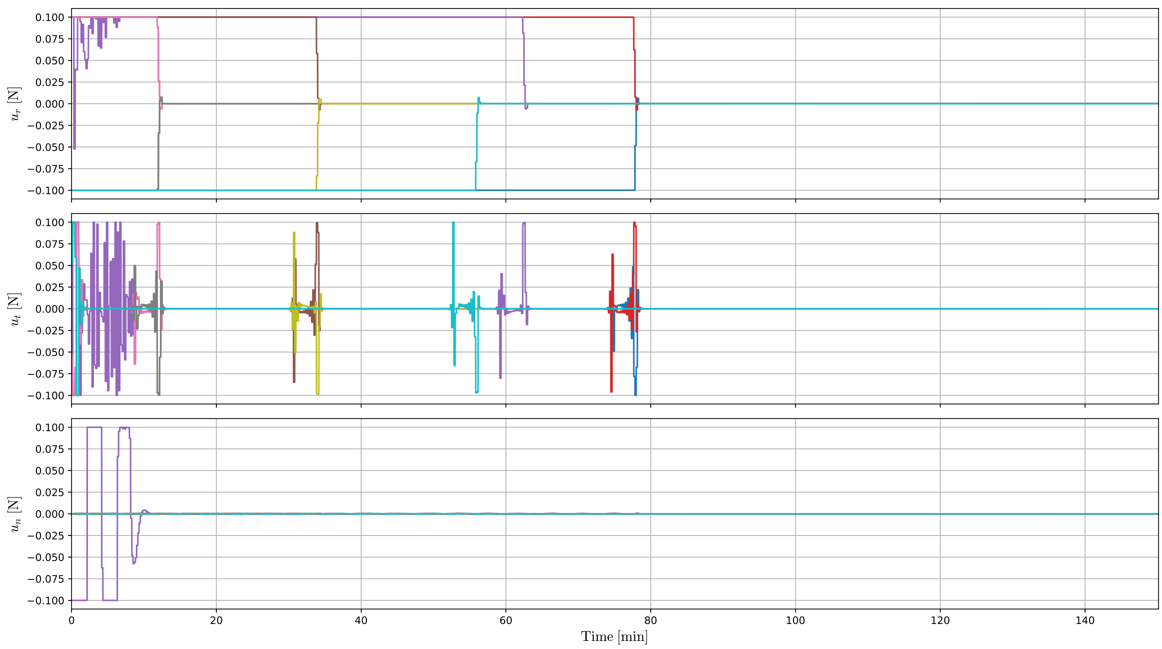


Figure F-8: Inputs for Hill-Clohessy-Wiltshire model in small inter-planar scenario.

F-2-2 Relative Orbital Elements Model

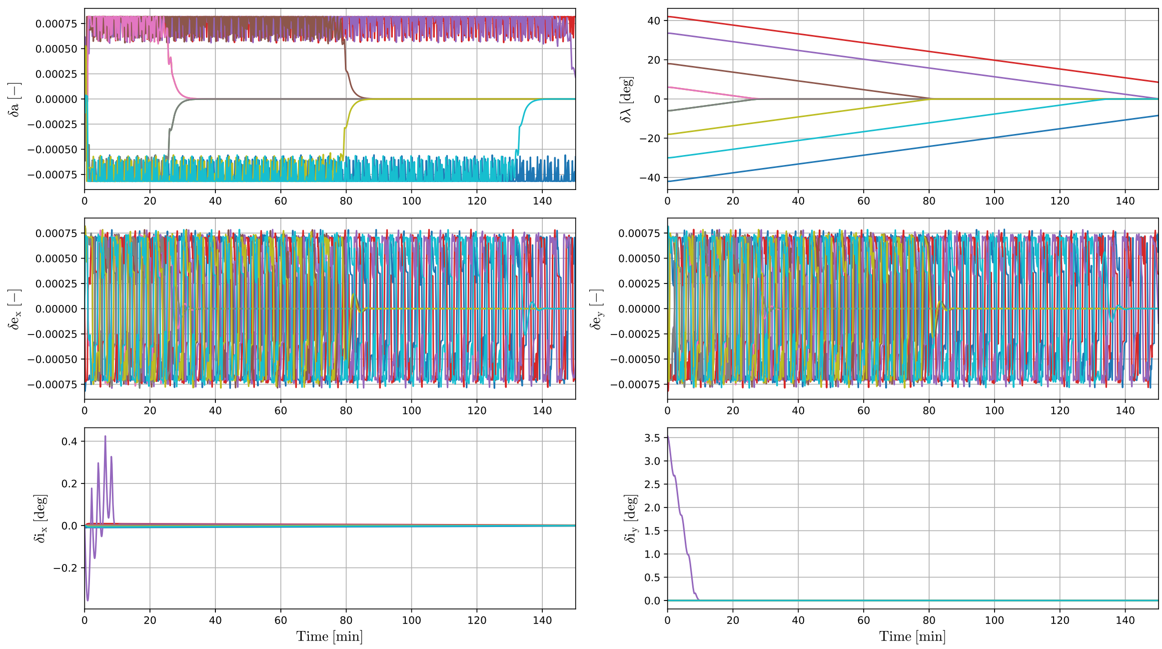


Figure F-9: States for Relative Orbital Elements model in small inter-planar scenario.

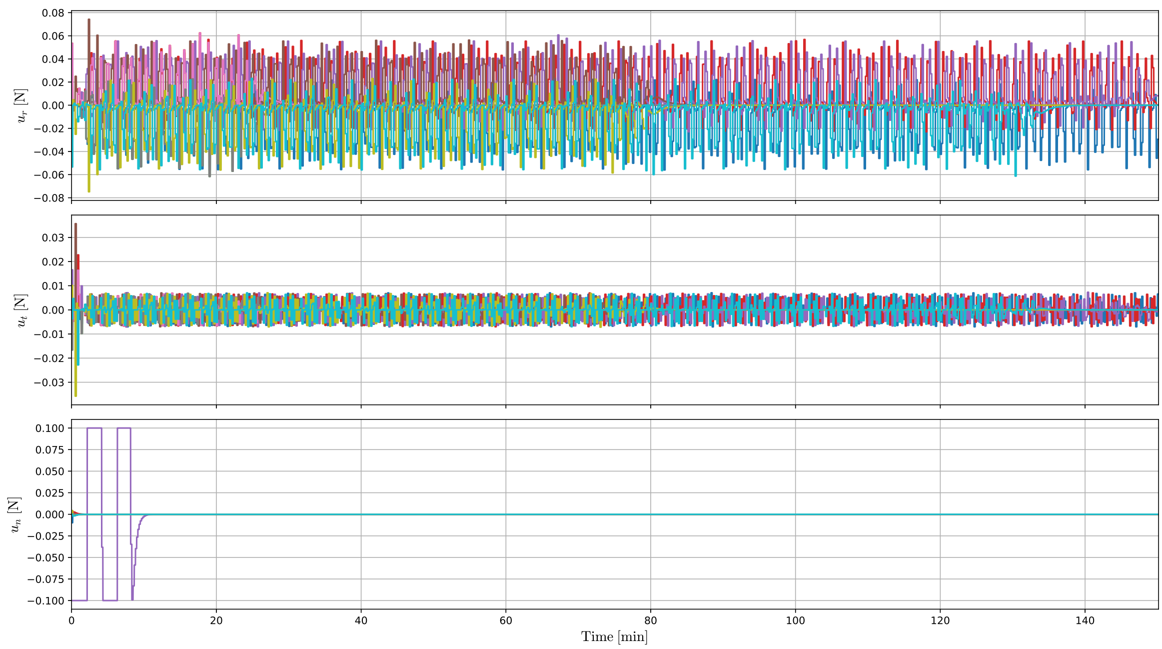


Figure F-10: Inputs for Relative Orbital Elements model in small inter-planar scenario.

F-2-3 Blend Model

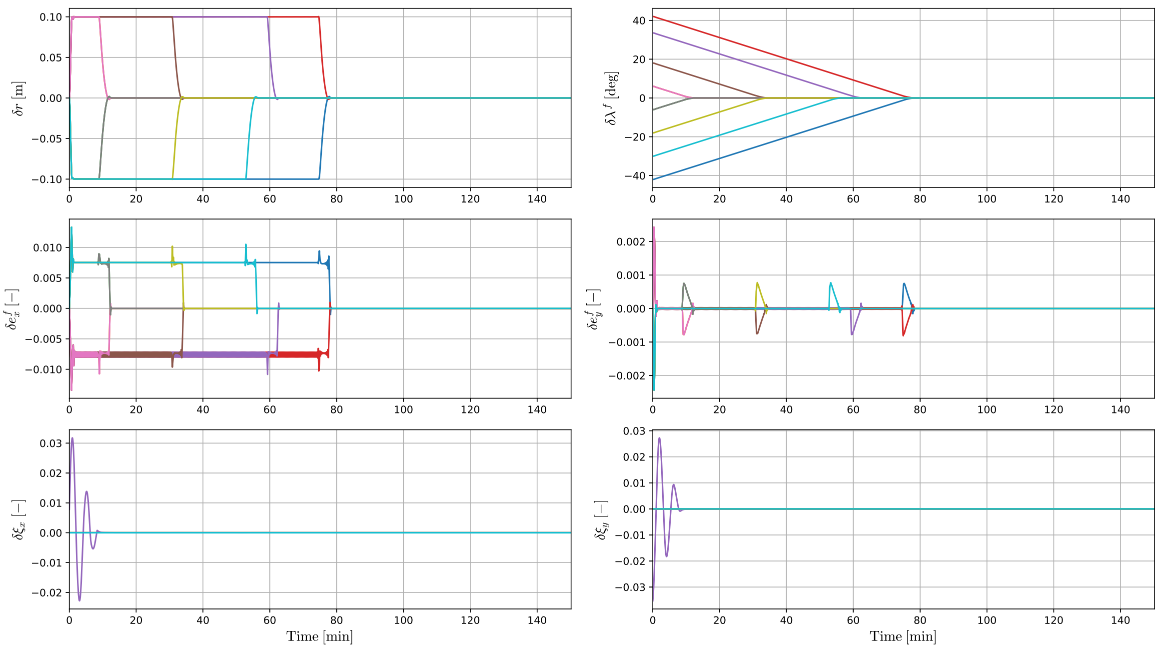


Figure F-11: States for Blend model in small inter-planar scenario.

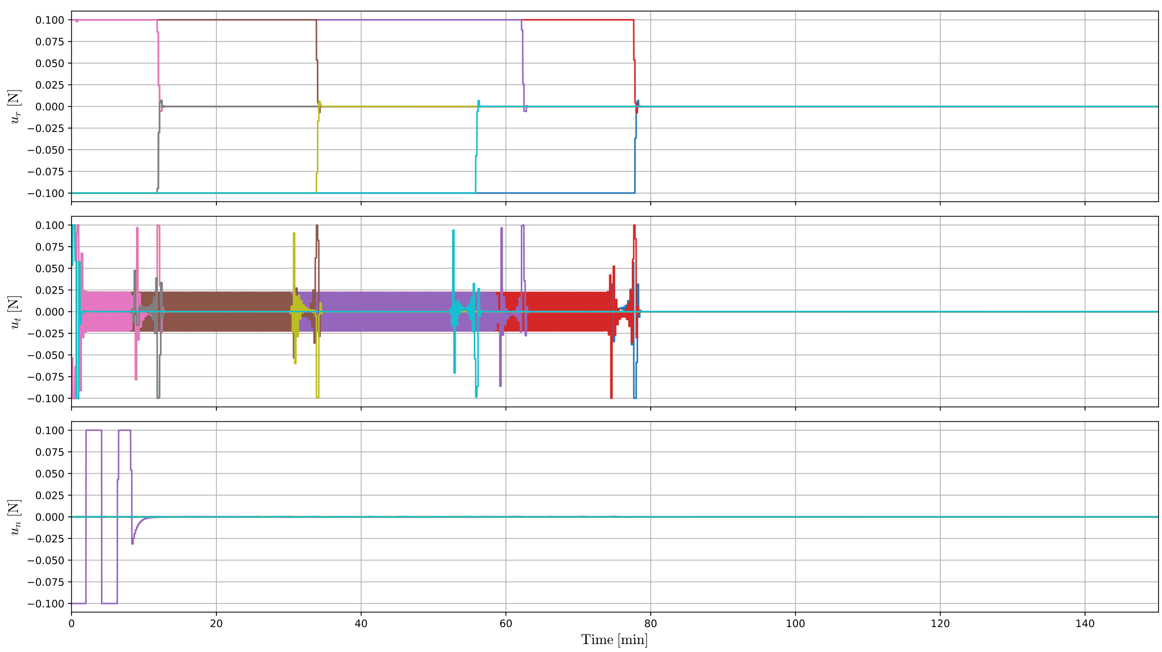


Figure F-12: Inputs for Blend model in small inter-planar scenario.

F-3 Model Comparison With Large Inter-Planar Simulation

F-3-1 Relative Orbital Elements Model

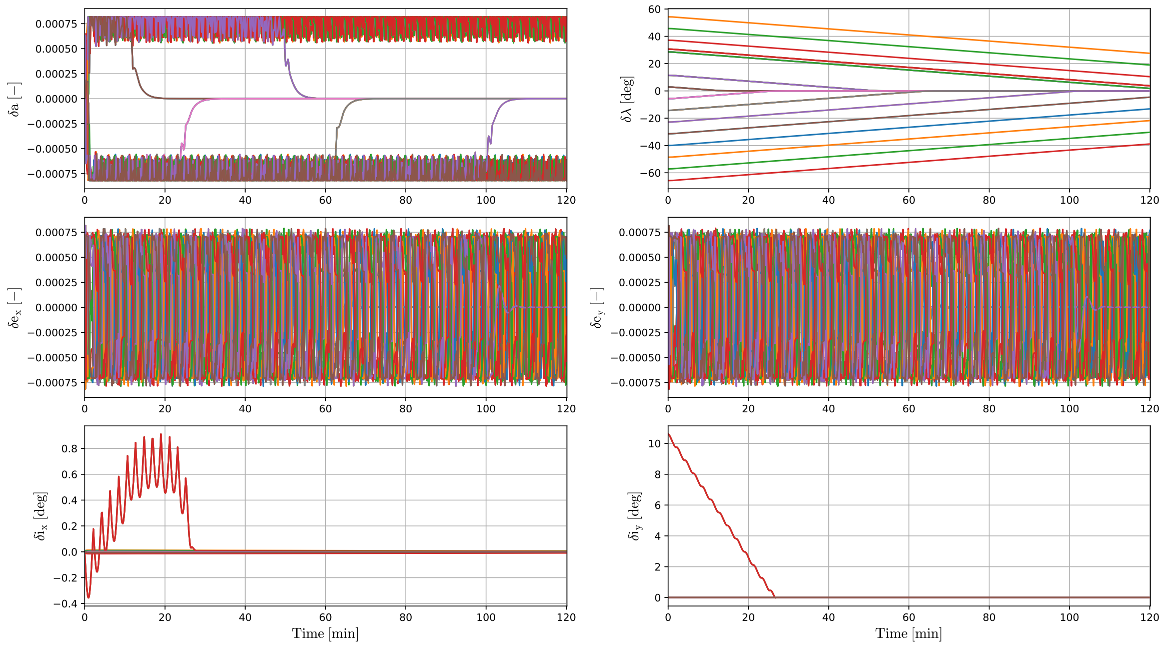


Figure F-13: States for Relative Orbital Elements model in large inter-planar scenario.

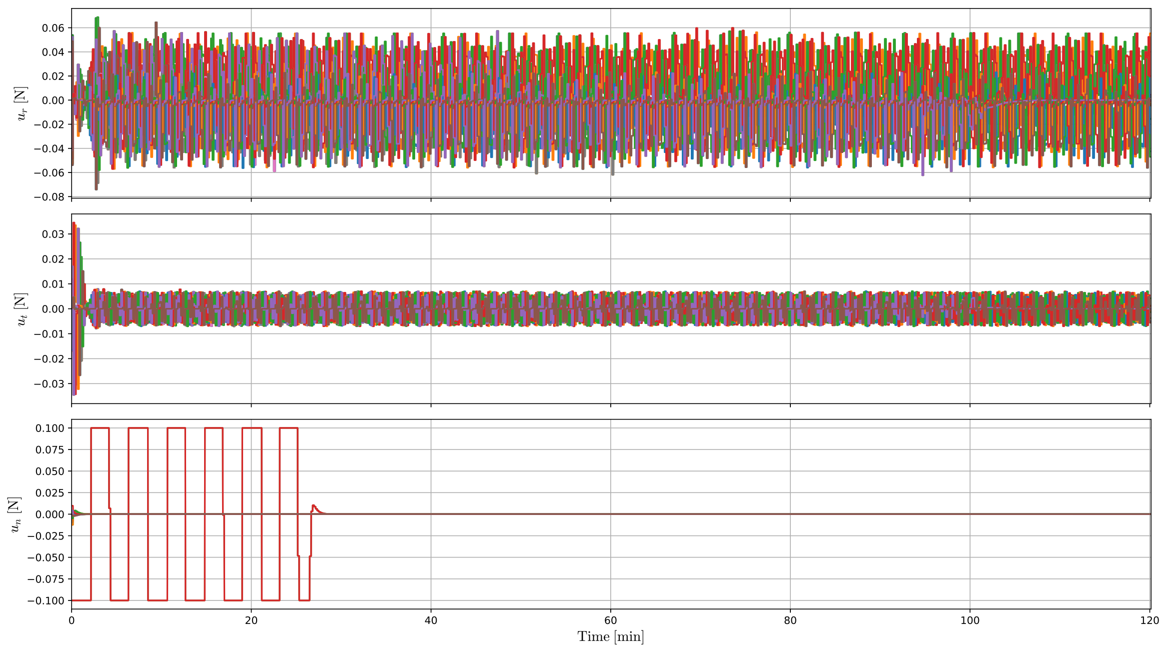


Figure F-14: Inputs for Relative Orbital Elements model in large inter-planar scenario.

F-3-2 Blend Model

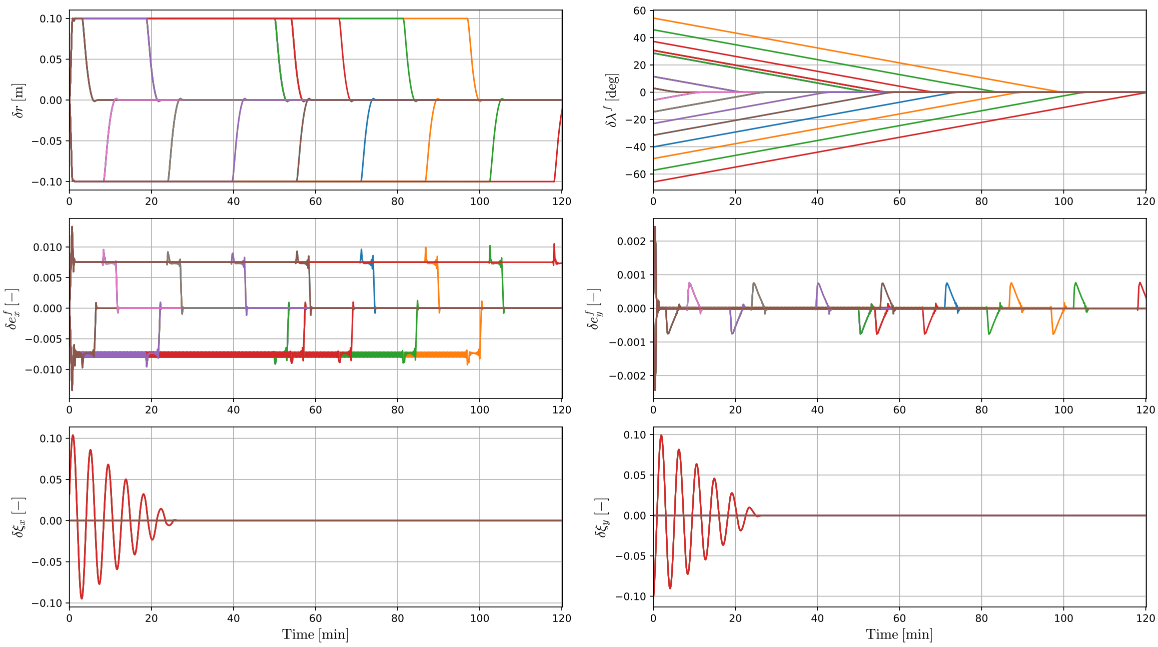


Figure F-15: States for Blend model in large inter-planar scenario.

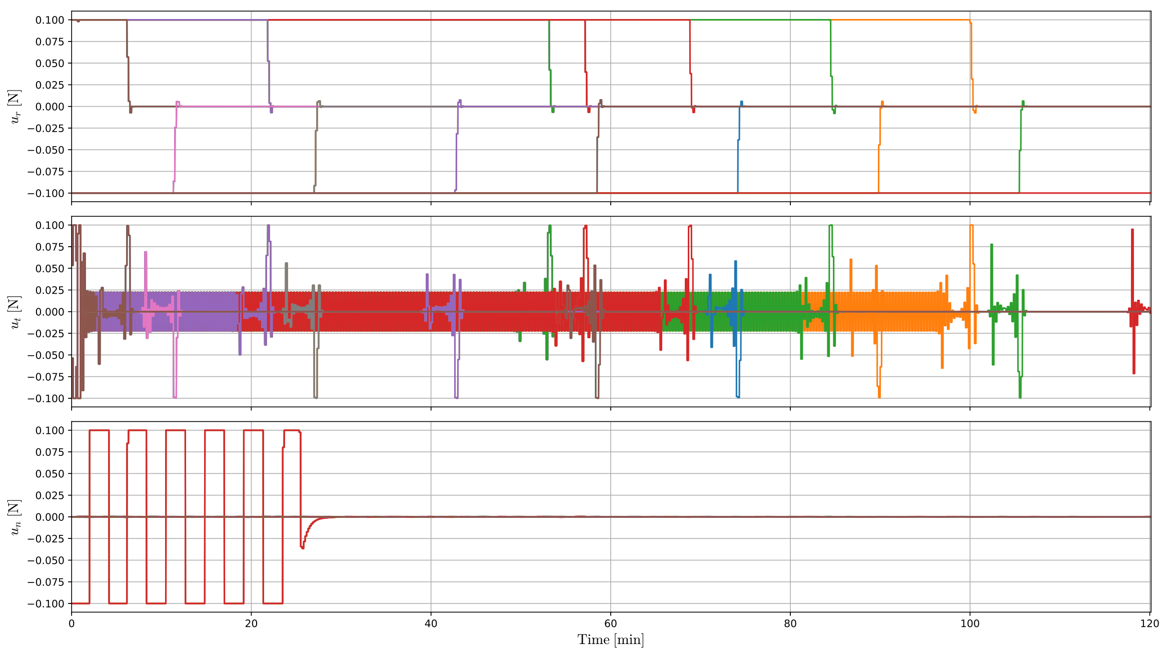


Figure F-16: Inputs for Blend model in large inter-planar scenario.

F-4 Robust SLS Comparison With Model Uncertainty Only

F-4-1 Nominal System-Level Synthesis

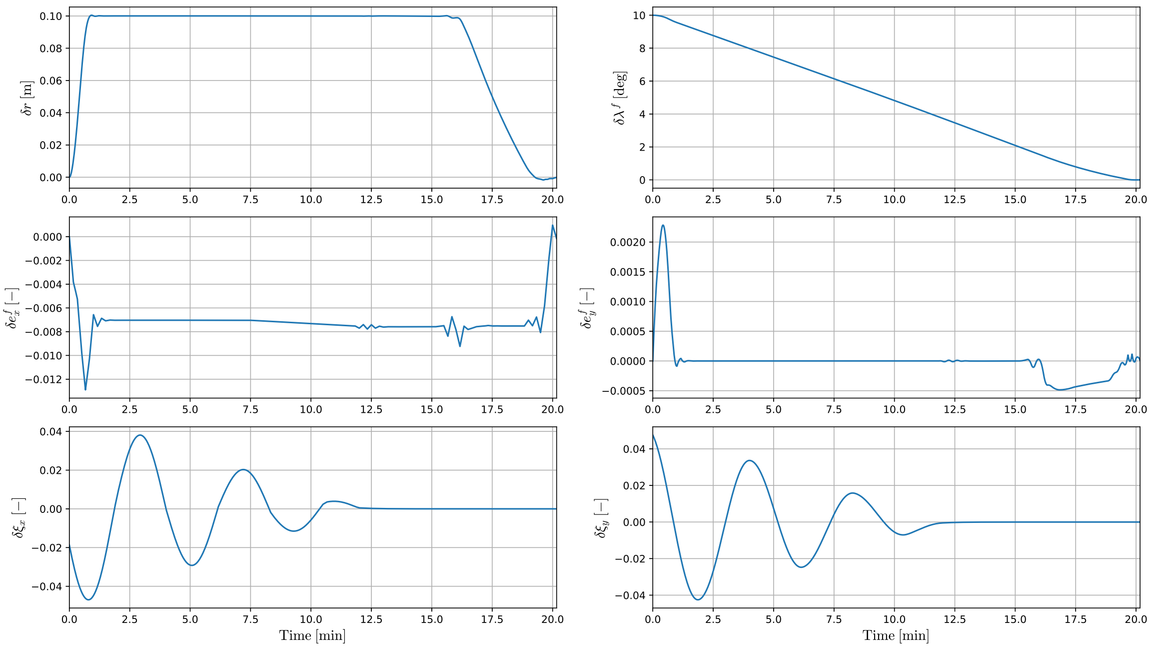


Figure F-17: States for nominal controller model with model uncertainty only.

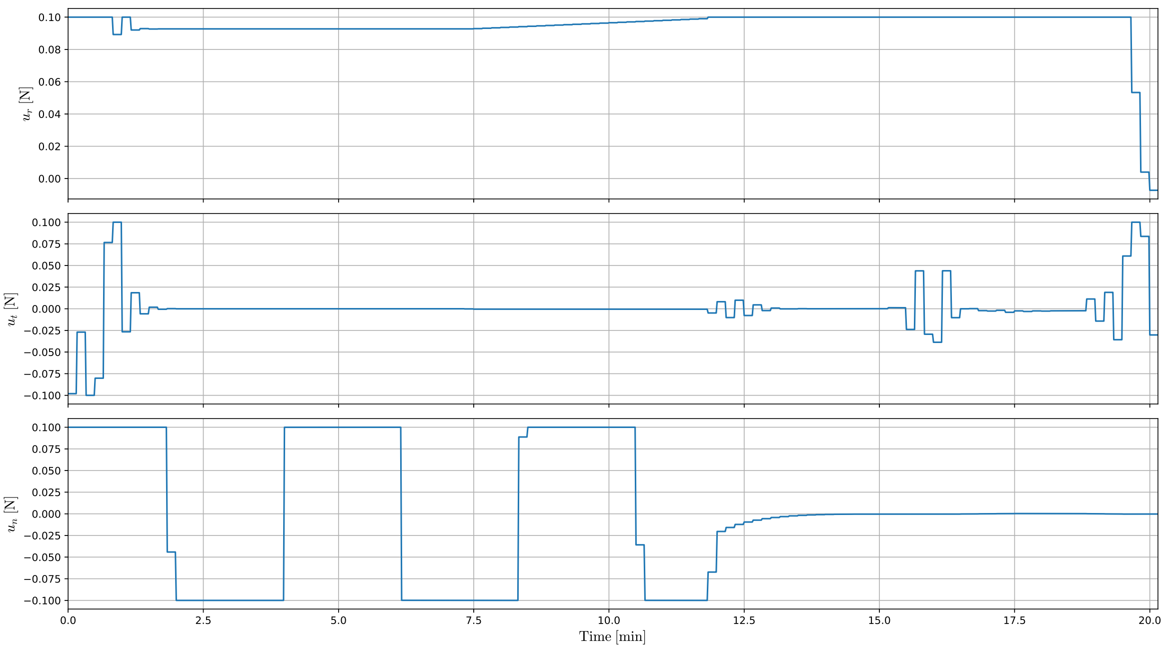


Figure F-18: Inputs for nominal controller model with model uncertainty only.

F-4-2 Classical Lumped System-Level Synthesis

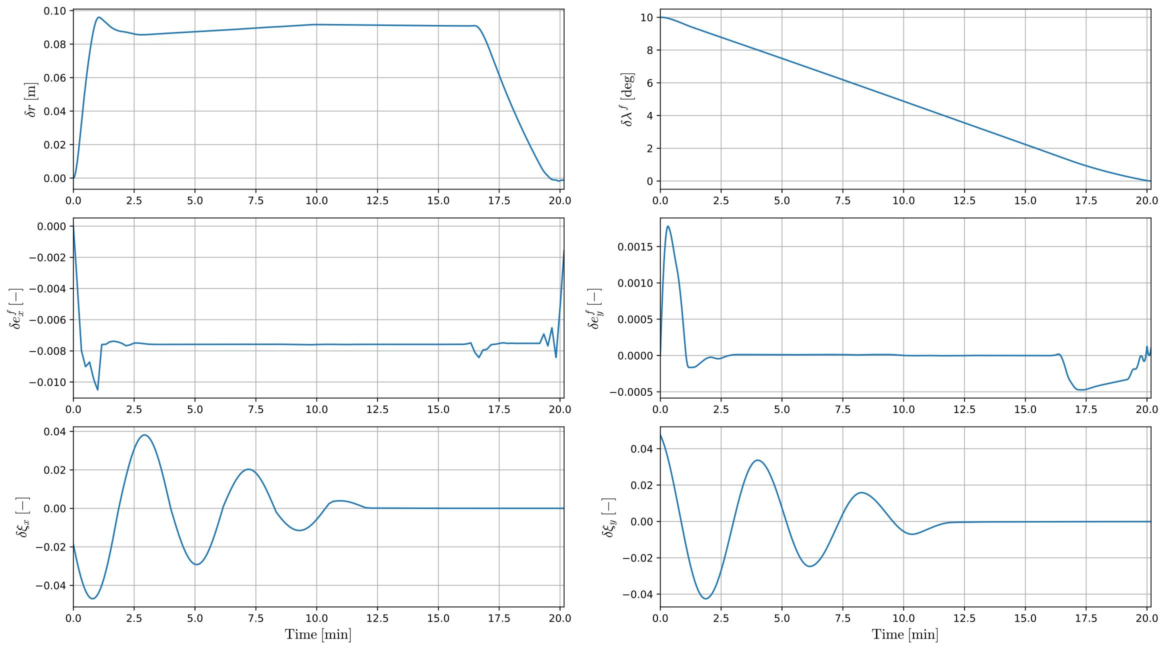


Figure F-19: States for CLSLS controller model with model uncertainty only.

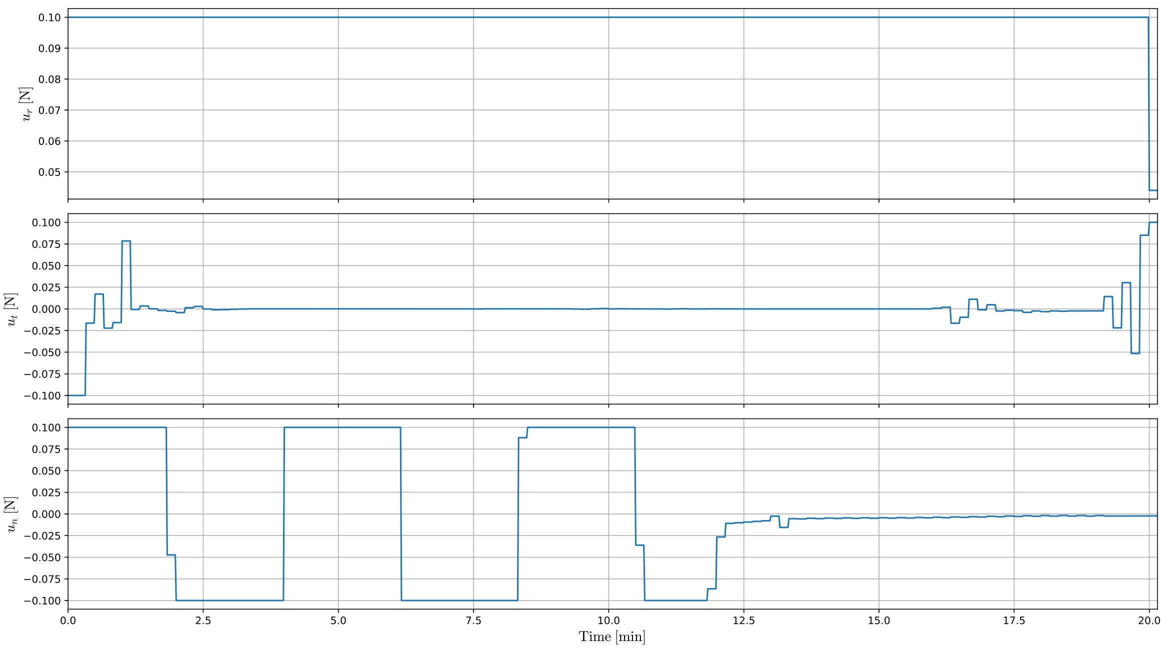


Figure F-20: Inputs for CLSLS controller model with model uncertainty only.

F-4-3 Improved Lumped System-Level Synthesis

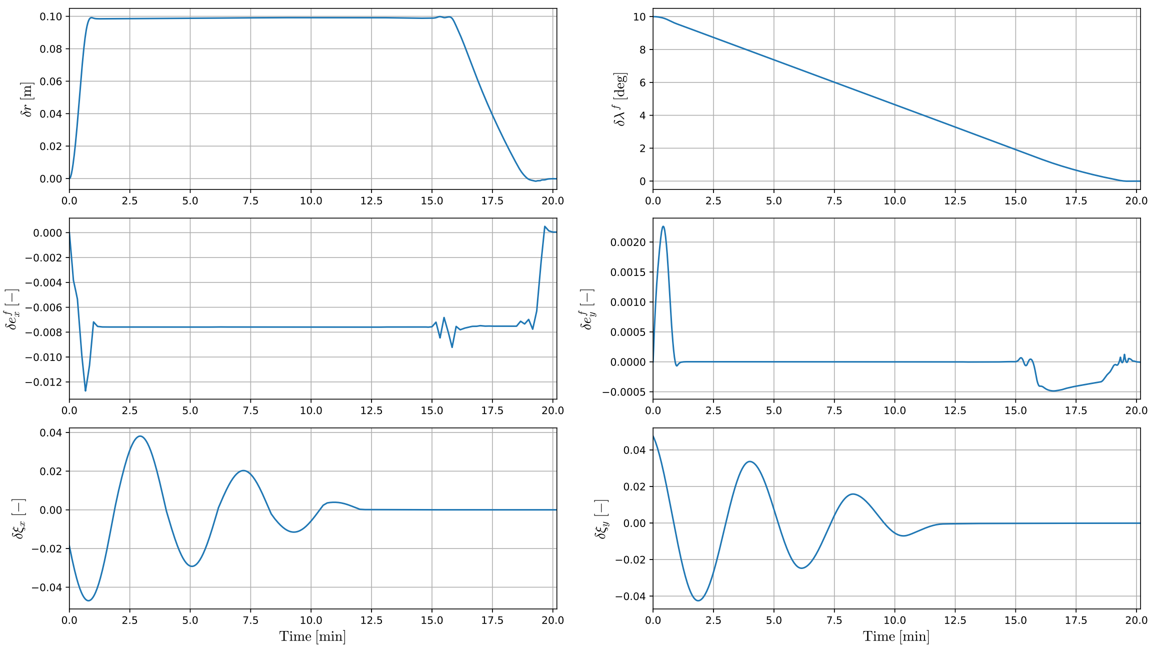


Figure F-21: States for ILSLS controller model with model uncertainty only.

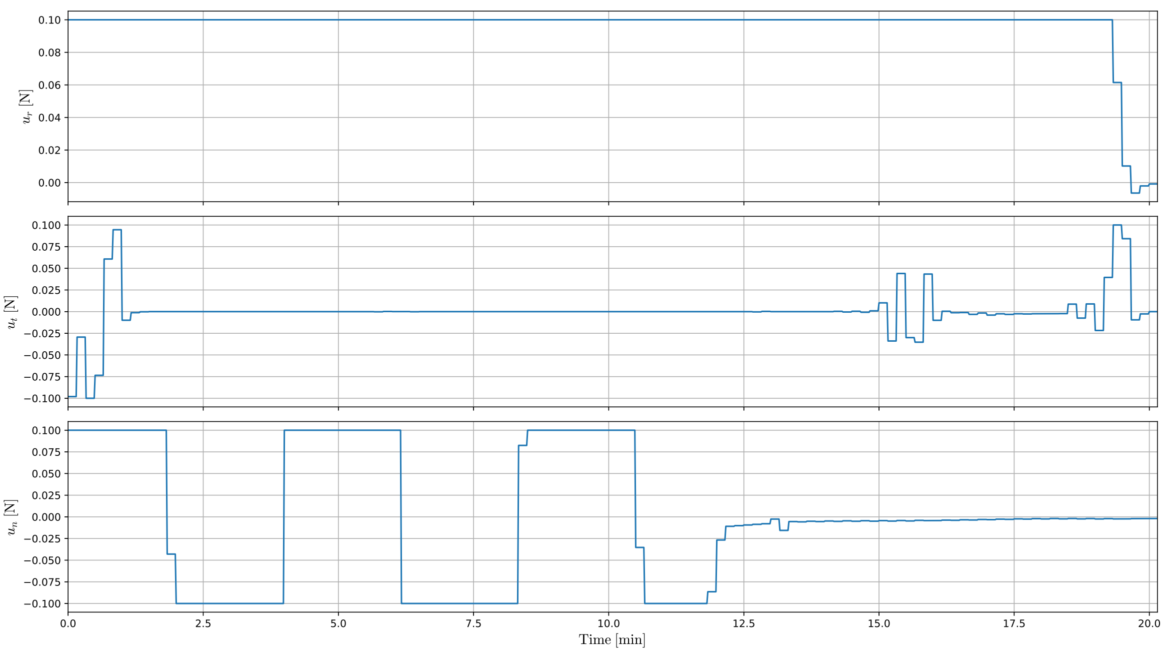


Figure F-22: Inputs for ILSLS controller model with model uncertainty only.

F-5 Robust SLS Comparison With Disturbances

F-5-1 Nominal System-Level Synthesis

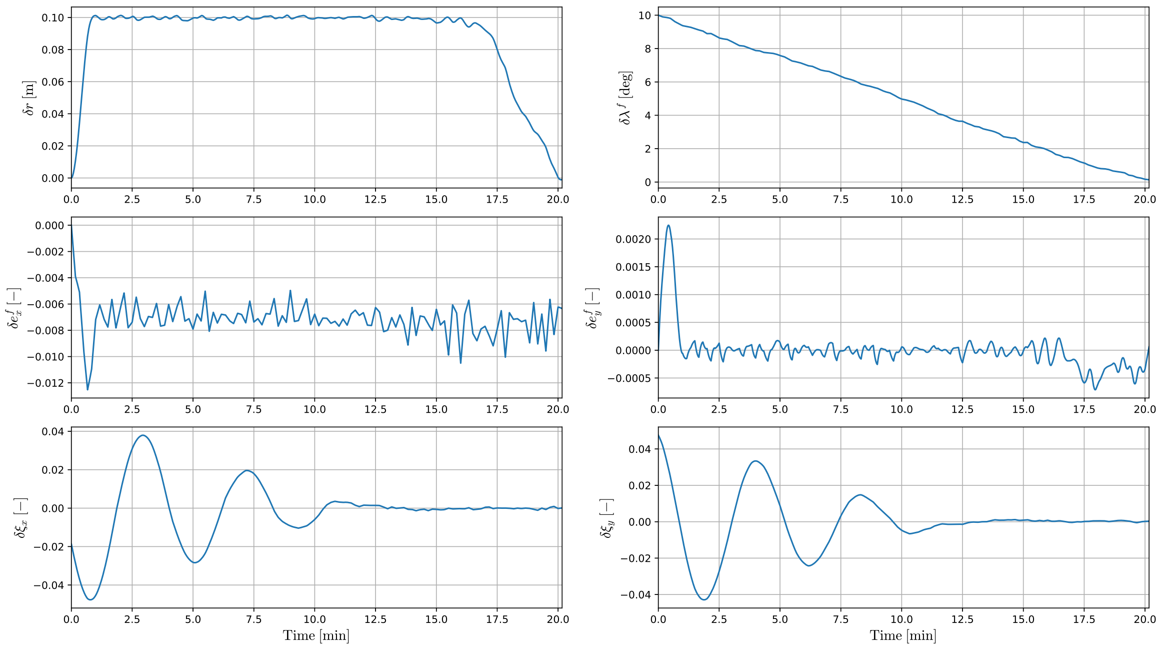


Figure F-23: States for nominal controller model with disturbances.

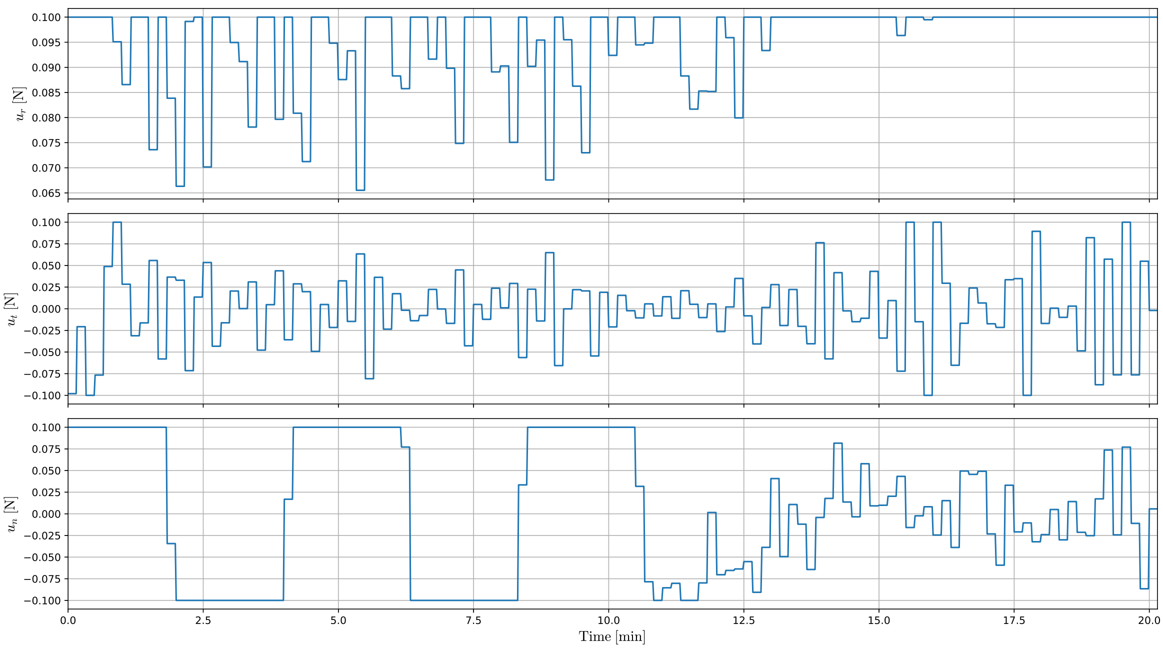


Figure F-24: Inputs for nominal controller model with disturbances.

F-5-2 Classical Lumped System-Level Synthesis

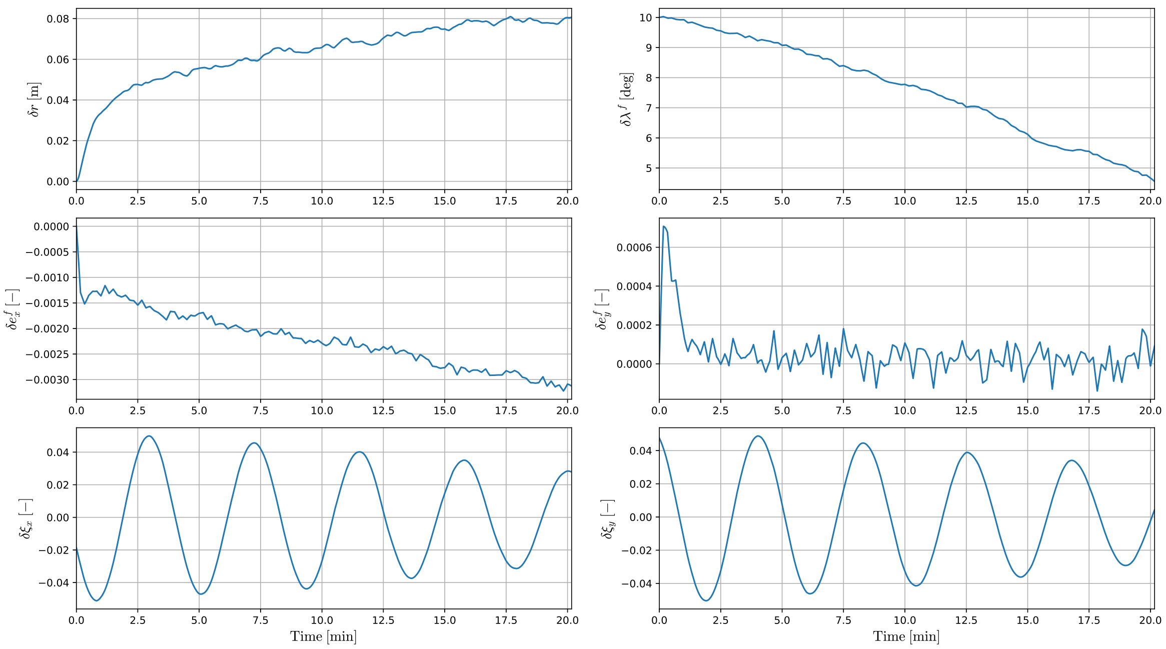


Figure F-25: States for CLSLS controller model with disturbances.

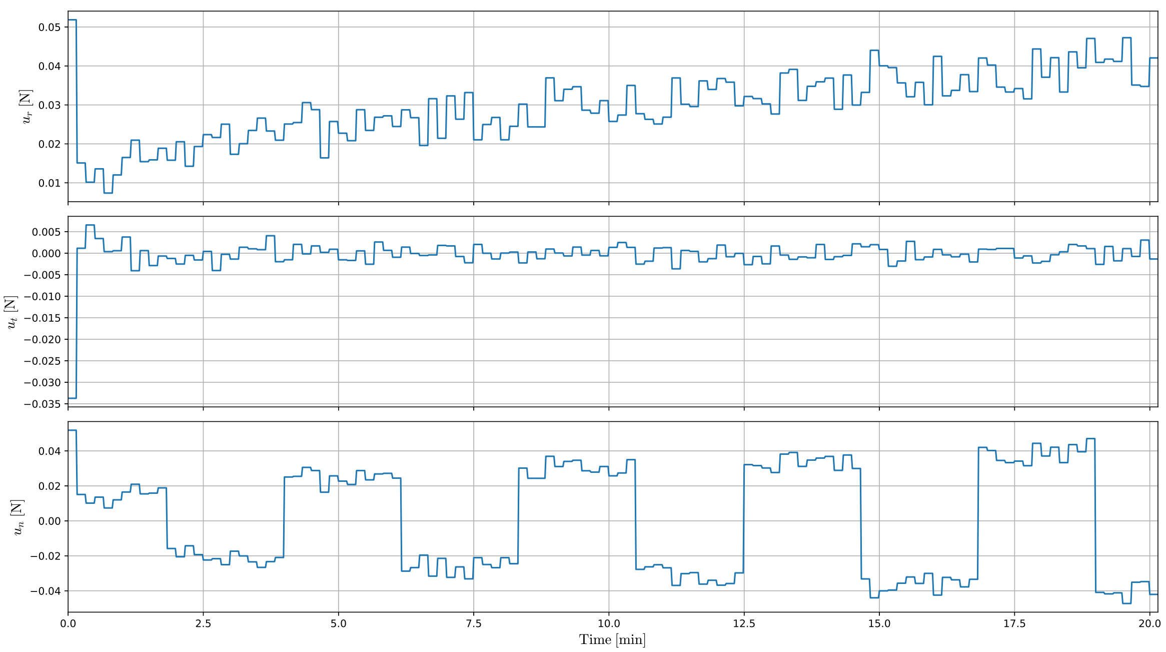


Figure F-26: Inputs for CLSLS controller model with disturbances.

F-5-3 Improved Lumped System-Level Synthesis

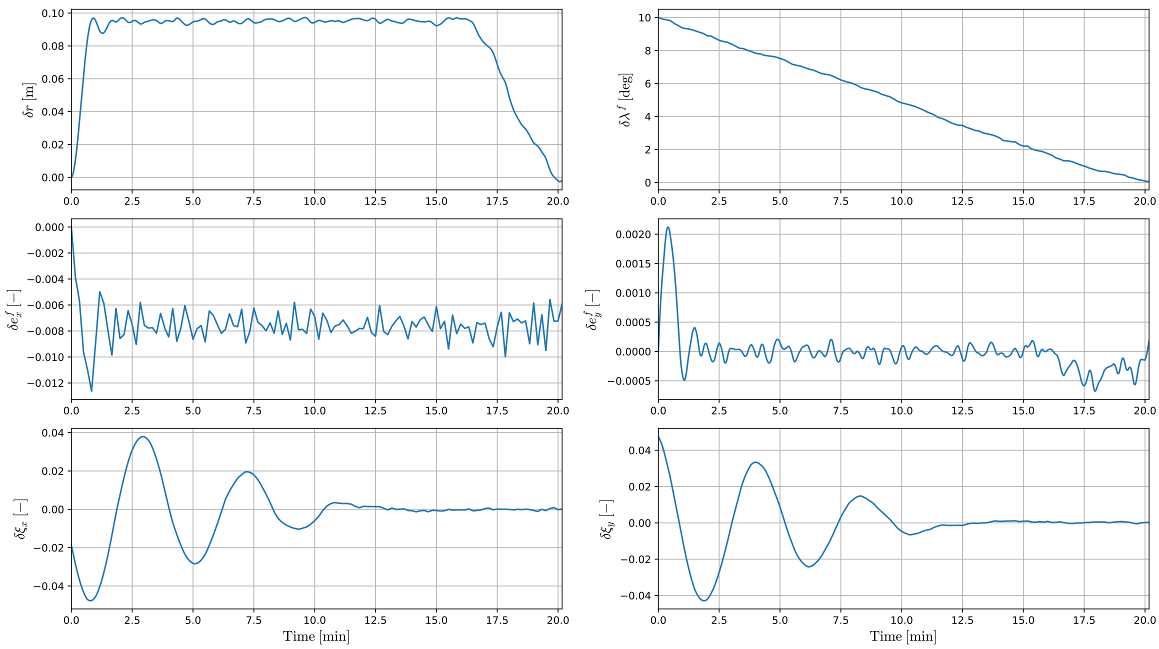


Figure F-27: States for ILSLS controller model with disturbances.

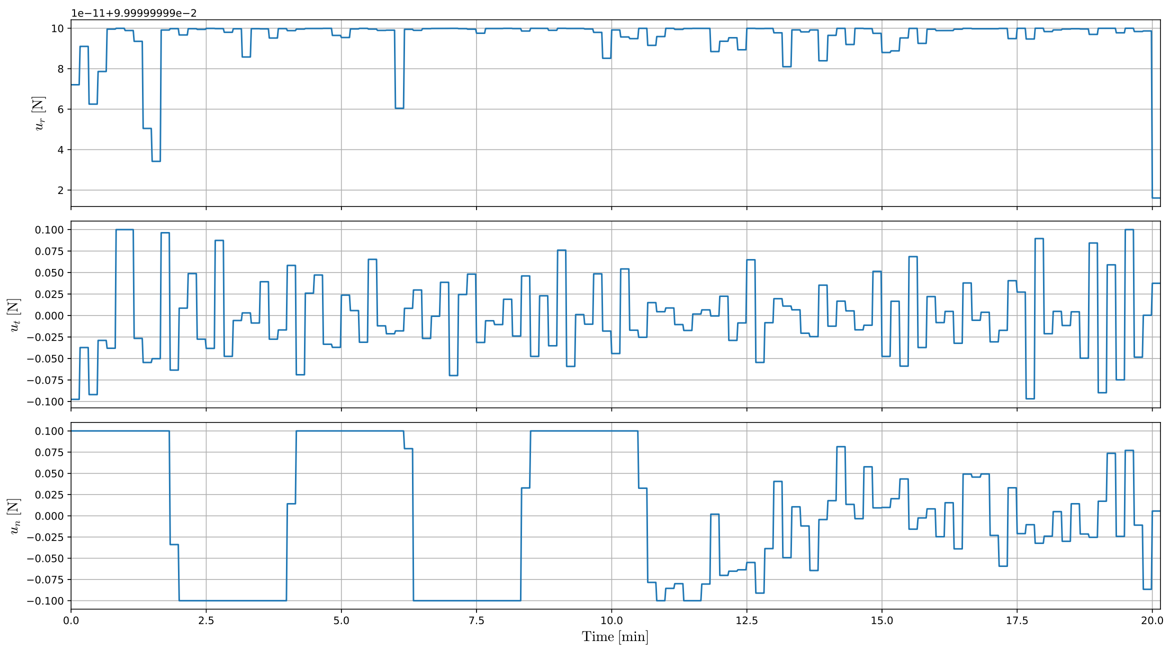


Figure F-28: Inputs for ILSLS controller model with disturbances.

F-6 Large Simulations

F-6-1 Nominal Controller Without Constraints And Disturbance

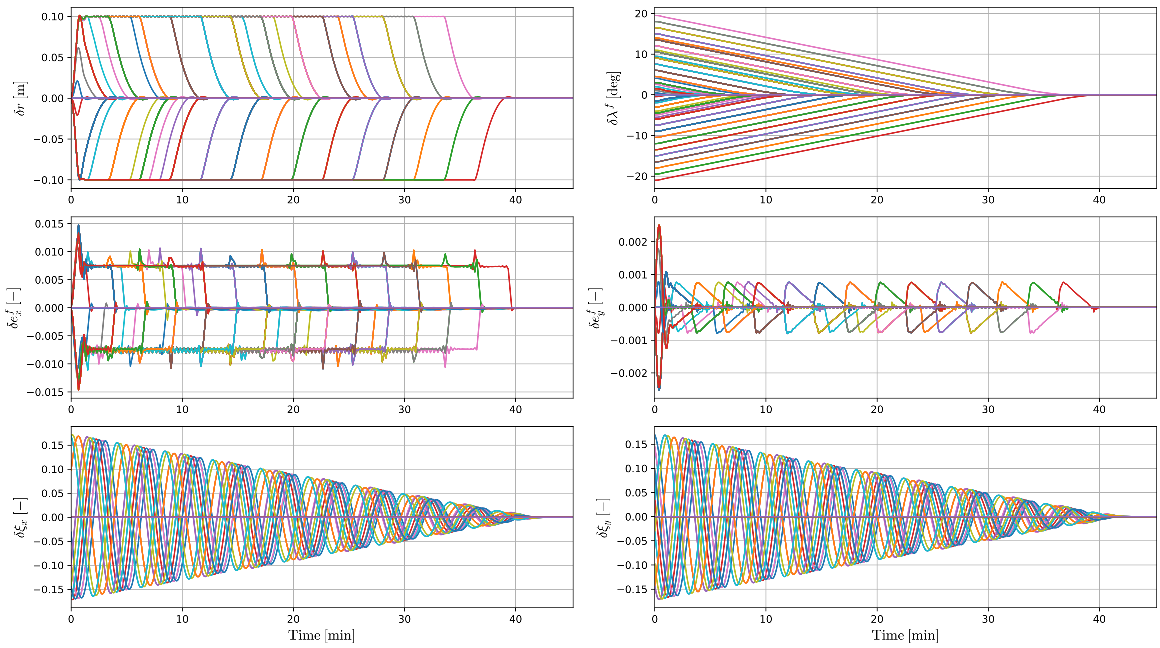


Figure F-29: States for nominal controller without constraints and disturbance.

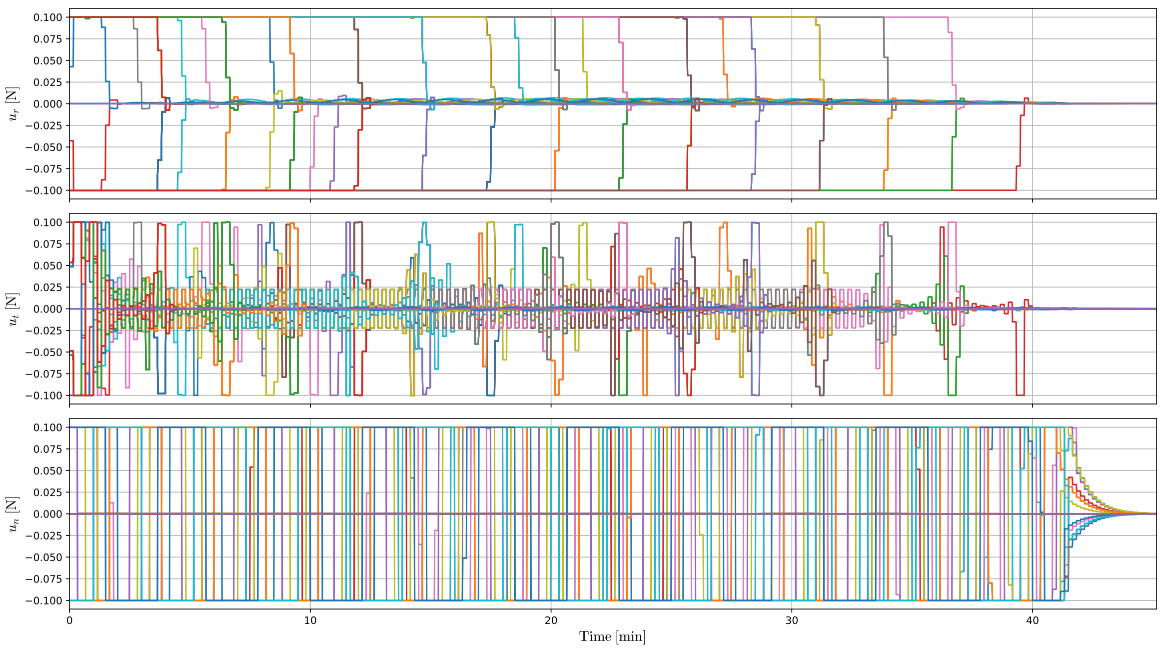


Figure F-30: Inputs for nominal controller without constraints and disturbance.

F-6-2 Nominal Controller With Constraints And Without Disturbance

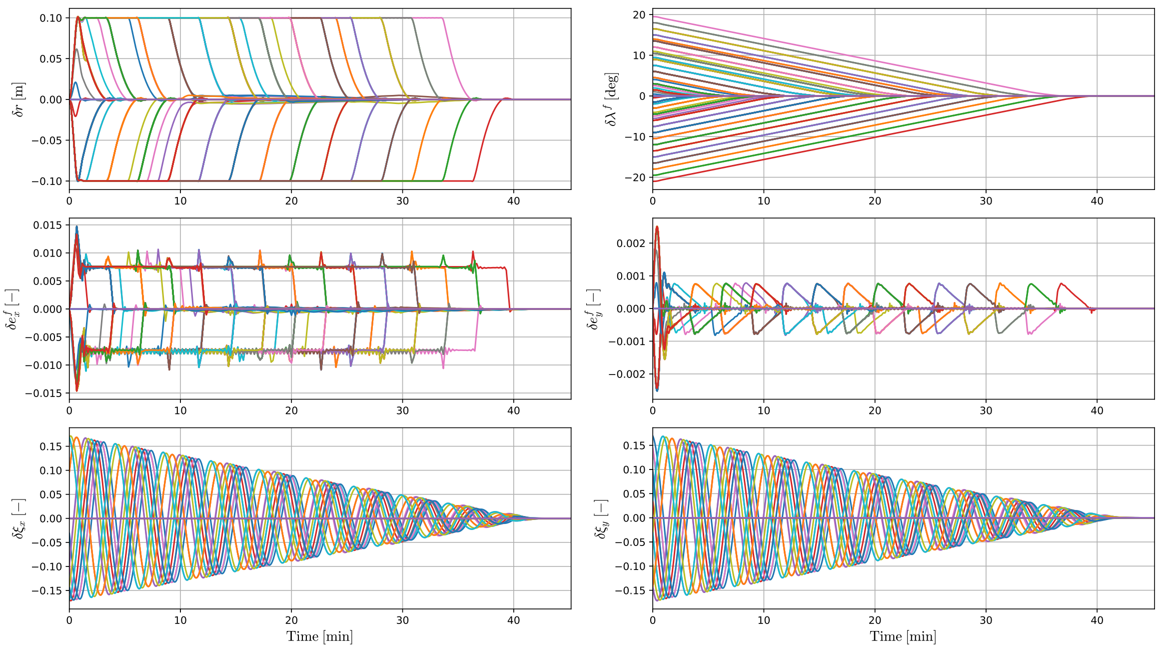


Figure F-31: States for nominal controller with constraints and without disturbance.

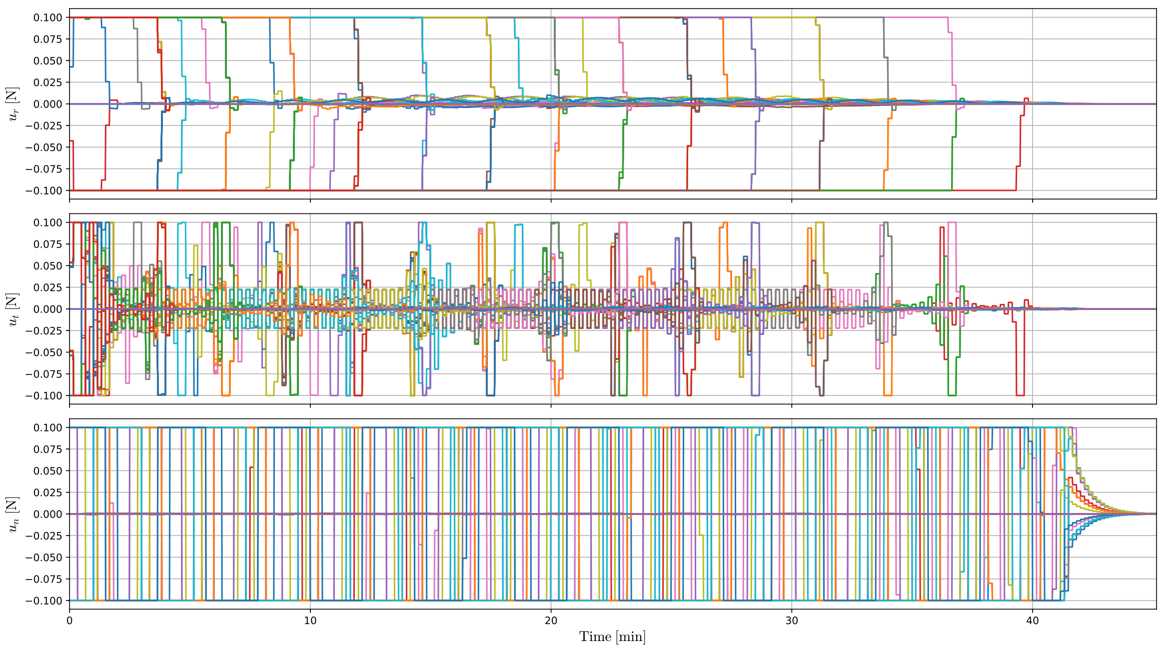


Figure F-32: Inputs for nominal controller with constraints and without disturbance.

F-6-3 Nominal Controller With Constraints And Disturbance

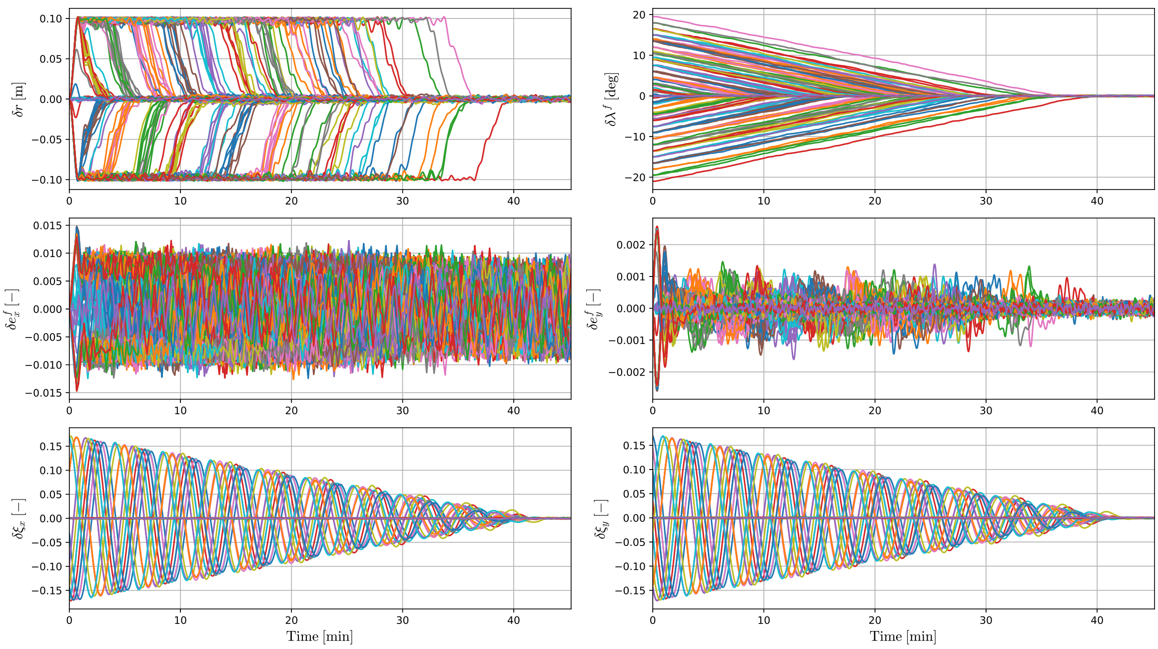


Figure F-33: States for nominal controller with constraints and disturbance.

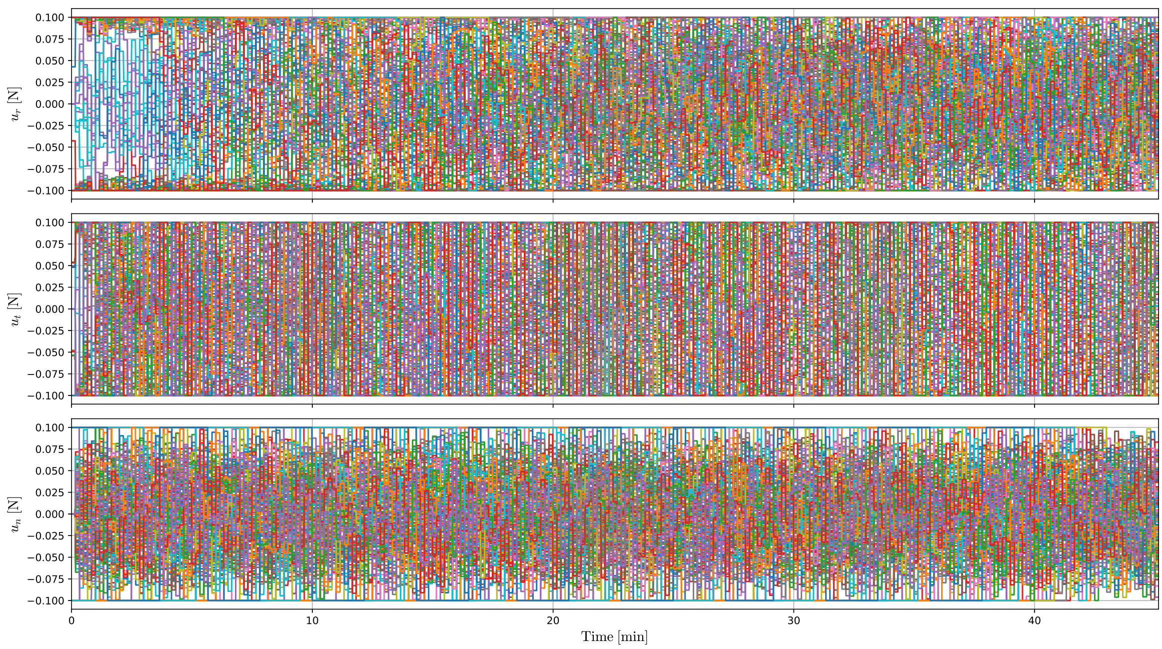


Figure F-34: Inputs for nominal controller with constraints and disturbance.

F-6-4 Robust Controller With Constraints And Disturbance

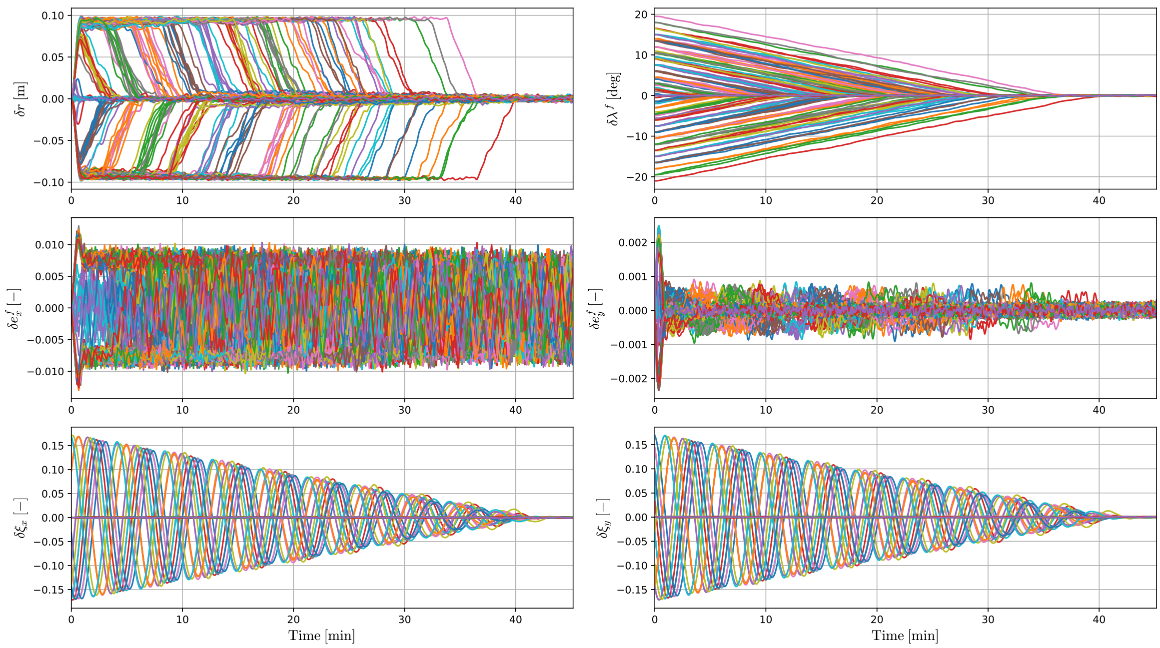


Figure F-35: States for robust controller with constraints and disturbance.

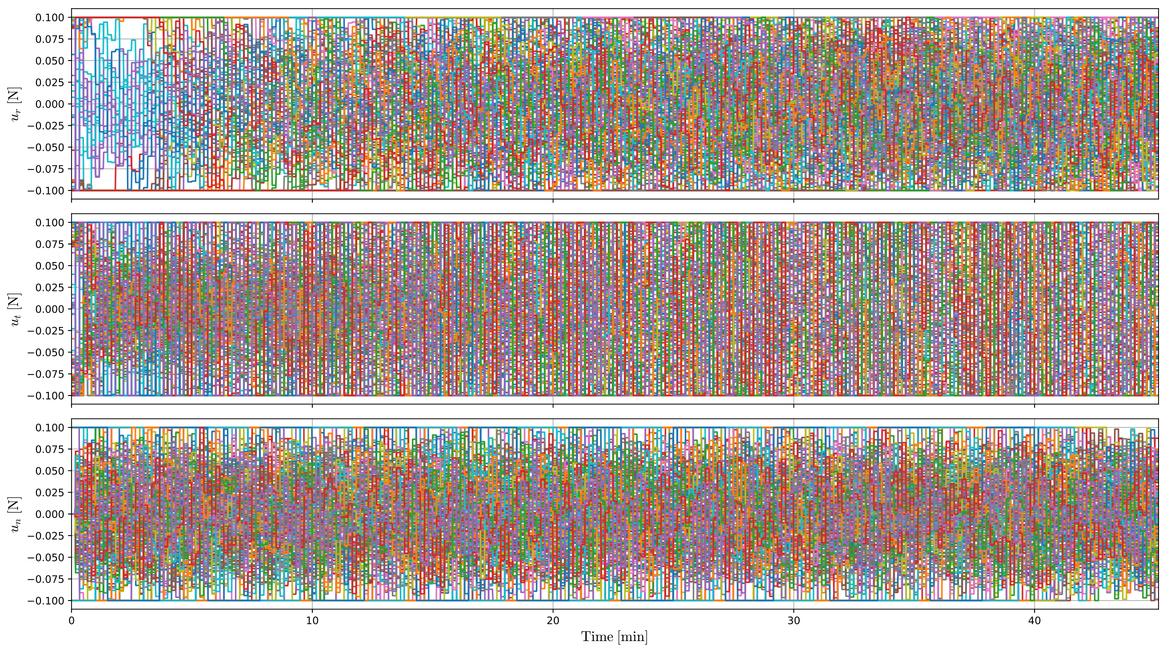


Figure F-36: Inputs for robust controller with constraints and disturbance.

Bibliography

[1] O. Kodheli, E. Lagunas, N. Maturo, *et al.*, "Satellite Communications in the New Space Era: A Survey and Future Challenges," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 1, pp. 70–109, Feb. 2021, ISSN: 1553877X. DOI: 10.1109/COMST.2020.3028247.

[2] M. K. Ben-Larbi, K. Flores Pozo, T. Haylok, *et al.*, "Towards the automated operations of large distributed satellite systems. Part 1: Review and paradigm shifts," *Advances in Space Research*, vol. 67, no. 11, pp. 3598–3619, Jun. 2021, ISSN: 18791948. DOI: 10.1016/j.asr.2020.08.009.

[3] C. Foster, H. Hallam, and J. Mason, "Orbit Determination and Differential-drag Control of Planet Labs Cubesat Constellations," Sep. 2015. [Online]. Available: http://arxiv.org/abs/1509.03270.

[4] E. Gill, "Together in Space: Potentials and Challenges of Distributed Space Systems," Delft University of Technology, Delft, Tech. Rep., Sep. 2008.

[5] Delft University of Technology, *Orbiting low frequency array (OLFAR)*. [Online]. Available: https://www.tudelft.nl/innovatie-impact/home-of-innovation/innovation-projects/orbiting-low-frequency-array-olfar.

[6] S. Engelen, C. Verhoeven, and l. M. Bentum, "OLFAR, A Radio Telescope Based On Nano-Satellites in Moon Orbit," in *24th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, USA: AIAA/ISU, Aug. 2010.

[7] E. Vos, J. M. Scherpen, and A. J. Van der Schaft, "Equal distribution of satellite constellations on circular target orbits," *Automatica*, vol. 50, no. 10, pp. 2641–2647, Oct. 2014, ISSN: 00051098. DOI: 10.1016/j.automatica.2014.08.027.

[8] R. Radhakrishnan, W. W. Edmonson, F. Afghah, R. M. Rodriguez-Osorio, F. Pinto, and S. C. Burleigh, "Survey of Inter-Satellite Communication for Small Satellite Systems: Physical Layer to Network Layer View," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 4, pp. 2442–2473, Oct. 2016, ISSN: 1553877X. DOI: 10.1109/COMST.2016.2564990.

[9] N. Pachler, I. del Portillo, E. F. Crawley, and B. G. Cameron, "An Updated Comparison of Four Low Earth Orbit Satellite Constellation Systems to Provide Global Broadband," in *IEEE International Conference on Communications*, Montréal, Canada: IEEE, Jun. 2021, ISBN: 9781728194417. DOI: 10.1109/ICCWorkshops50388.2021.9473799.

[10] B. Su and Q. Zhou, "Analysis of Dynamic Evolution and Station-Keeping of Starlink," in *China Automation Congress (CAC)*, Xiamen, China: IEEE, Jun. 2021, pp. 8229–8234, ISBN: 9781665426473. DOI: 10.1109/CAC53003.2021.9728207.

[11] C. Foster, J. Mason, V. Vittaldev, *et al.*, "Constellation Phasing with Differential Drag on Planet Labs Satellites," *Journal of Spacecraft and Rockets*, vol. 55, no. 2, pp. 473–483, Mar. 2018, ISSN: 15336794. DOI: 10.2514/1.A33927.

[12] C. Foster, J. Mason, V. Vittaldev, *et al.*, "Differential Drag Control Scheme for Large Constellation of Planet Satellites and On-Orbit Results," Jun. 2018. [Online]. Available: http://arxiv.org/abs/1806.01218.

[13] A. Li and J. Mason, "Optimal Utility of Satellite Constellation Separation with Differential Drag," in *AIAA/AAS Astrodynamics Specialist Conference*, San Diego, CA, USA: AIAA, Aug. 2014, ISBN: 9781624103087. DOI: 10.2514/6.2014-4112.

[14] M. K. Ben-Larbi, K. Flores Pozo, M. Choi, *et al.*, "Towards the automated operations of large distributed satellite systems. Part 2: Classifications and tools," *Advances in Space Research*, vol. 67, no. 11, pp. 3620–3637, Jun. 2021, ISSN: 18791948. DOI: 10.1016/j.asr.2020.08.018.

[15] K. T. Alfriend, S. R. Vadali, P. Gurfil, J. P. How, and L. S. Breger, *Spacecraft Formation Flying: Dynamics, control and navigation*. Elsevier/Butterworth-Heinemann, 2010, ISBN: 9780750685337.

[16] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*. American Institute of Aeronautics \& Astronautics, 1999, ISBN: 1600860265.

[17] D. Brouwer, "Solution of the Problem of Artificial Satellite Theory Without Drag," *The Astronomical Journal*, vol. 64, pp. 378–396, Nov. 1959.

[18] K. T. Alfriend and H. Yan, "Evaluation and Comparison of Relative Motion Theories," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 2, pp. 254–261, Mar. 2005, ISSN: 15333884. DOI: 10.2514/1.6691.

[19] R. Bate, D. D. Mueller, and J. E. White, *Fundamentals of Astrodynamics*. Dover Publications, 1971, pp. 421–422, ISBN: 0486600610.

[20] J. E. Prussing and B. A. Conway, *Orbital Mechanics*. Oxford University Press, Inc., 1993, pp. 155–169, ISBN: 0-19-507834-9.

[21] G. W. Hill, "Researches in the Lunar Theory," *American Journal of Mathematics*, vol. 1, no. 1, pp. 5–26, 1878. [Online]. Available: https://www.jstor.org/stable/2369430.

[22] W. H. Clohessy and R. S. Wiltshire, "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences*, vol. 27, pp. 653–658, Sep. 1960. DOI: https://doi.org/10.2514/8.8704.

[23] D. K. Geller and T. A. Lovell, "Angles-Only Initial Relative Orbit Determination Performance Analysis using Cylindrical Coordinates," *Journal of the Astronautical Sciences*, vol. 64, no. 1, pp. 72–96, Mar. 2017, ISSN: 21950571. DOI: 10.1007/s40295-016-0095-z.

[24] T. Pippia, V. Preda, S. Bennani, and T. Keviczky, "Reconfiguration of a satellite constellation in circular formation orbit with decentralized model predictive control," *arXiv*, Jan. 2022. [Online]. Available: http://arxiv.org/abs/2201.10399.

[25] J. Sullivan, S. Grimberg, and S. D'Amico, "Comprehensive Survey and Assessment of Spacecraft Relative Motion Dynamics Models," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 8, pp. 1837–1859, 2017, ISSN: 15333884. DOI: 10.2514/1.G002309.

[26] W. Langson, I. Chryssochoos, S. V. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, Jan. 2004, ISSN: 0005-1098. DOI: 10.1016/J.AUTOMATICA.2003.08.009.

[27] M. Bujarbaruah, U. Rosolia, Y. R. Stürz, and F. Borrelli, “A Simple Robust MPC for Linear Systems with Parametric and Additive Uncertainty,” Mar. 2021. [Online]. Available: http://arxiv.org/abs/2103.12351.

[28] S. Chen, N. Matni, M. Morari, and V. M. Preciado, “System Level Synthesis-based Robust Model Predictive Control through Convex Inner Approximation,” *arXiv*, Nov. 2021. [Online]. Available: http://arxiv.org/abs/2111.05509.

[29] J. Anderson, J. C. Doyle, S. Low, and N. Matni, “System Level Synthesis,” *Annual Reviews in Control*, vol. 47, pp. 364–393, Apr. 2019. [Online]. Available: http://arxiv.org/abs/1904.01634.

[30] A. Xue and N. Matni, “Data-Driven System Level Synthesis,” *Proceedings of Machine Learning Research*, vol. 144, pp. 1–12, 2021.

[31] C. A. Alonso, J. S. Li, N. Matni, and J. Anderson, “Distributed and Localized Model Predictive Control. Part II: Theoretical Guarantees,” *arXiv*, Mar. 2022. [Online]. Available: http://arxiv.org/abs/2203.00780.

[32] J. S. Li and J. C. Doyle, “Distributed Robust Control for Systems with Structured Uncertainties,” *arXiv*, Apr. 2022. [Online]. Available: http://arxiv.org/abs/2204.02493.

[33] S. Chen, H. Wang, M. Morari, V. M. Preciado, and N. Matni, “Robust Closed-loop Model Predictive Control via System Level Synthesis,” in *IEEE Conference on Decision and Control (CDC)*, Jeju Island, Republic of Korea: IEEE, Dec. 2020, pp. 2152–2159, ISBN: 9781728174471. DOI: 10.1109/CDC42340.2020.9304200.

[34] Y. S. Wang, N. Matni, and J. C. Doyle, “Separable and Localized System-Level Synthesis for Large-Scale Systems,” *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4234–4249, Dec. 2018, ISSN: 15582523. DOI: 10.1109/TAC.2018.2819246.

[35] A. P. Leeman, J. Sieber, S. Bennani, and M. N. Zeilinger, “Robust Optimal Control for Nonlinear Systems with Parametric Uncertainties via System Level Synthesis,” Apr. 2023. [Online]. Available: http://arxiv.org/abs/2304.00752.

[36] S. Chen, V. M. Preciado, M. Morari, and N. Matni, “Robust Model Predictive Control with Polytopic Model Uncertainty through System Level Synthesis,” Mar. 2022. [Online]. Available: http://arxiv.org/abs/2203.11375.

[37] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, “OSQP: an operator splitting solver for quadratic programs,” *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, Dec. 2020, ISSN: 18672957. DOI: 10.1007/s12532-020-00179-2.

[38] M. Schubiger, G. Banjac, and J. Lygeros, “GPU acceleration of ADMM for large-scale quadratic programming,” *Journal of Parallel and Distributed Computing*, vol. 144, pp. 55–67, Oct. 2020, ISSN: 07437315. DOI: 10.1016/j.jpdc.2020.05.021.

[39] S.-H. Tseng and J. S. Li, “SLSpy: Python-Based System-Level Controller Synthesis Framework,” Apr. 2020. [Online]. Available: http://arxiv.org/abs/2004.12565.

[40] G. Gaias, J. S. Ardaens, and O. Montenbruck, “Model of J2 perturbed satellite relative motion with time-varying differential drag,” *Celestial Mechanics and Dynamical Astronomy*, vol. 123, no. 4, pp. 411–433, Aug. 2015, ISSN: 15729478. DOI: 10.1007/s10569-015-9643-2.

[41] M. Khalil, B. Larbi, and E. Stoll, "Spacecraft Formation Control Using Analytical Integration Of Gauss' Variational Equations," Tech. Rep.

[42] M. Leomanni, G. Bianchini, A. Garulli, and A. Giannitrapani, "State Feedback Control in Equinoctial Variables for Orbit Phasing Applications," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 8, Aug. 2018.

Index