12-2024

Finite-Time Control Strategies for Rendezvous and Proximity Operations

John Tamotsu Akagi
Utah State University

FINITE-TIME CONTROL STRATEGIES FOR RENDEZVOUS AND PROXIMITY OPERATIONS

by

John Tamotsu Akagi

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Aerospace Engineering

Approved:

Matthew W. Harris, Ph.D. Major Professor	Tianyi He, Ph.D. Committee Member
Douglas Hunsaker, Ph.D. Committee Member	Greg Droge, Ph.D. Committee Member
Burak Sarsılmaz, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2024

ABSTRACT

Finite-Time Control Strategies for Rendezvous and Proximity Operations

by

John Tamotsu Akagi, Doctor of Philosophy

Utah State University, 2024

Major Professor: Matthew W. Harris, Ph.D.
Department: Mechanical and Aerospace Engineering

Finite-time control laws guarantee a system will converge to an equilibrium point at some finite time, but often lack the ability to incorporate additional objectives. The goal of this dissertation is to develop strategies for finite-time control so that the convergence time, state, and control bounds can be considered while maintaining the convergence guarantees of the finite-time controller. Four different control approaches are examined in this dissertation: model predictive control, common Lyapunov functions, backstepping, and switching surfaces. For these approaches, the relationship between the tuning parameters and the convergence time, state errors, and control usage is examined and then used to design control laws that incorporate these objectives. For some approaches, explicit bounds on control usage or convergence time are able to be determined while for other approaches, optimization problems are developed which guide system behavior while still maintaining the finite-time guarantees. For control laws which rely on a switching function, the hyperbolic tangent function is used to approximate the function and minimize control chatter. The scaling factor used with the hyperbolic tangent is included as a tunable parameter and is tuned to minimize chatter while still maintaining the desired system properties. The tuned control laws are tested and verified in simulation using the spacecraft rendezvous system, with both the simplified Clohessy-Wiltshire dynamics and the two-body nonlinear

dynamics. The results show that the proper tuning of these finite-time control laws does allow for control over these secondary objectives, and strategies are developed to guide the tuning of each law.

(191 pages)

PUBLIC ABSTRACT

Finite-Time Control Strategies for Rendezvous and Proximity Operations

John Tamotsu Akagi

One common metric to use when designing a controller is the time that the system will take to reach the desired state. Unfortunately, many approaches to developing controllers only guarantee that the system will approach, but not exactly reach, the desired state. This can become a limitation in time-sensitive situations where rapid and complete convergence is necessary. One group of control methods, known as finite-time control, does guarantee both faster convergence and that the desired state will be reached, but often fails to define exactly what time that will occur, how much control will be used to get there, and how much error the system will experience before arriving. This dissertation examines a selection of control laws which guarantee finite-time stability. Each control law has parameters which can be selected, and the relationship between those parameters and the controller performance are analyzed. This allows for easier and faster design of controllers which guarantee finite-time convergence and also allows for other system behaviors to be designed for. The tuned control laws were tested and validated in simulations which modeled the rendezvous between two spacecraft.

ACKNOWLEDGMENTS

I am extremely grateful to my advisor, Dr. Matt Harris, for the countless ways he's assisted, mentored, taught, and guided me over the course of my graduate studies. I have learned much from him through both coursework and as I've worked on this dissertation, and I could not have completed it without his patience, support, advice, and insights.

I would also like to thank the Space Dynamics Laboratory for supporting this work through their PhD Tomorrow Fellowship. Additionally, I am grateful to Tyson and the whole Guidance, Navigation, and Control group at SDL for their mentoring.

Finally, I am forever grateful to my family, and especially my wife Courtney, for their support and encouragement.

John Akagi

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

When designing a controller for a system, various competing factors must be considered so that the controller can meet design objectives while respecting computational and physical constraints. For example, a spacecraft controller may leverage a simple, low-fidelity model because the onboard computer lacks the computational power available to an Earth-based computer. The controller may need to avoid saturating the actuators, preserve safe distances between spacecraft, and complete tasks within an allotted time. A common trade-off in control design is between control usage and system responsiveness. The challenge in designing a controller is then balancing the different constraints and trade-offs such that the overall objectives can be met. The goal of this dissertation is to develop strategies for finite-time control so that the convergence time, state, and control bounds can be considered while maintaining the convergence guarantees of the finite-time controller.

1.1 Stability Classifications

A controller's primary goals are to stabilize a system operating point and drive the system to that operating point. One common classification of controllers is based on the time required to converge to the desired state. When the system approaches the desired state in the limit as time goes to infinity, it is known as asymptotic convergence. When there is some time when the system exactly reaches the desired state, it is known as finite-time convergence. When the controller also stabilizes the operating point, the points are then asymptotically stable and finite-time stable, respectively. Each of these classifications can be further sub-divided based on the mathematical guarantees available.

1.1.1 Asymptotic Stability

When the origin of a system is asymptotically stable, the state approaches zero in the limit as time goes to infinity. As such, the system does not have a well-defined time at which it "arrives" at the desired state. One approach to quantify how fast the system converges is to measure how long it takes to reach certain benchmarks. Two common ones are the rise time, which is the time it takes to reach some percentage of the final value, and the settling time, which is when the system approaches and stays within some band of the final value [1]. However, there are not universally agreed upon ranges for these and they can differ between engineers.

Another approach is to bound the system response by a known function. This is the approach to define exponential stability where the error function is bounded by an exponentially decaying function. While the error still only approaches zero asymptotically, exponential stability does guarantee a minimum rate for the decrease in error.

Despite the lack of a definitive convergence time, asymptotic control remains useful and the most prevalent form of control in the literature. For example, state-space techniques, such as pole placement and linear quadratic regulation, for control of linear time-invariant systems asymptotically stabilize the origin. Additionally, the control input is simply a linear combination of the system states so the control laws are simple to implement and execute quickly. This may be satisfactory when having an easily implemented, stabilizing controller is sufficient or when computing resources are limited.

1.1.2 Finite-Time Stability

In contrast to asymptotic stability, finite-time stability guarantees that the system under consideration converges to the desired state in finite time. The relationship between the initial conditions of the system and the convergence time is known as the settling-time function. The relationship between the controller's tuning parameters, the settling-time function, and other performance metrics is often difficult to quantify. Thus, the design of

finite-time controllers may prove difficult in practice. Addressing this difficulty is a key element of this dissertation.

Sub-classifications of finite-time stability are available based on properties of the settling-time function. Fixed-time controllers are finite-time stable but also have a known upper bound on the settling-time function which allows for a known, worst-case convergence time. Predefined-time controllers are a further subcategory where the upper bound for the settling-time function is an explicit tuning parameter in the control law. Finally, prescibed-time controllers have been developed which allow for precise selection of the settling time.

These additional refinements allow for a greater degree of understanding and control over the convergence time bound. However, with the exception of prescribed-time controllers, this is subtly different from controlling the actual convergence time since the bound could be conservative to the point that it does not actually provide any useful guidance. Additionally, due to the focus on quantifying the convergence time, these controllers generally do not give any weight to other considerations such as state or control constraints.

1.2 Finite-Time Approaches

There is no singular approach to designing finite-time controllers and each approach has its own strengths, limitations, and applications. In general, these approaches build on other asymptotically-stabilizing, nonlinear techniques but add additional elements that provide the finite-time guarantees.

In [2–5], backstepping is used to develop the controller. Backstepping is an approach where each individual state in a system is stabilized as if it were a subsystem with the input being the following state. By repeating the process for each state, while maintaining the stability requirements for each lower state, the overall system can be stabilized about the origin. In addition to backstepping, [3] includes a tuning function which is used to set the convergence time, resulting in a predefined-time controller, while [4,5] use adaptive elements to estimate unknown parameters.

Input-to-state stability is a control approach where the magnitude of system states are bounded by functions of the states, time, and control magnitude. By adjusting the

requirements on the bounding function so that it goes to zero in finite-time, the input-to-state stability approach can be used to guarantee finite-time stability [6]. This approach was also used in [7], to generate fixed-time controllers, and [8] which examined impulsive systems.

Another nonlinear control approach is based on the control changing as the system reaches or crosses some set of designated surfaces. These arise naturally in minimum-time problems [9,10] which results in the "bang-bang" control where the controls have a constant, maximum magnitude but change directions as surfaces are reached. A related concept is sliding mode control, where the controller forces the system to a surface which is known to converge to the desired state. While in general, the convergence to the surface is finite-time, the convergence along the surface is asymptotic. However, by modifying the surface, the convergence along the surface can also be finite-time. This results in terminal sliding mode control which was developed and explored in [11–13].

Prescribed-time control laws have their roots in proportional navigation control laws which use time-varying control gains which approach infinity as the terminal time is reached. The first instance of a specific prescribed-time control law was presented in [14] which applied a time-varying control gain to the system and then designed a feedback law to stabilize the new system. This was shown to be successful at stabilizing the system with a bounded control when the system states were perfectly known.

One approach to prescribed-time control laws is to use a nonlinear function that goes to infinity as the prescribed time is approach and then transform it to a function which acts over an infinite time horizon. By controlling the performance of the transformed system, the original system can then be made to achieve prescribed-time convergence. This approach, known as time scaling, was introduced in [15] and expanded on in work such as [16] which used a "generalized finite-time gain function" to convert a baseline, non-finite time control law to a prescribed-time one. A more in-depth overview of prescribed-time control can be found in [17] which details approaches and applications for prescribed-time control

in single-input-single output systems, multi-input-multi-output systems, and distributed control systems.

1.3 Finite-Time Applications

Finite-time controllers have been used in a variety of settings where the convergence guarantees are desirable. In [18], a fixed-time controller is used with an autonomous vehicle to bound the convergence time needed to obtain a desired trajectory, even in the presence of disturbances or actuator faults. Similarly, in [19], a finite-time controller was developed for an autonomous vehicle which used a fuzzy logic controller to address disturbances in the system. For spacecraft applications, a sliding surface based, fixed-time controller was developed for a spacecraft rendezvous scenario in [20] where the surface could be tuned to obtain the desired convergence time. A spacecraft attitude control system was considered in [21], which guaranteed predefined-time stability, and [22], which developed a predefined-time controller for precision Mars landing missions. A list of additional finite-time applications includes aircraft aerial refueling [23], output feedback tracking [24], chemical processes [25, 26], as well as many others referenced in [27].

For this work, a spacecraft rendezvous scenario is used as the model for which the finite-time controllers are developed. Spacecraft rendezvous has been explored since the earliest days of spaceflight [28,29] and been used in notable applications such as the Moon landings, construction and operation of the International Space Station, and repair missions to the Hubble Space Telescope. These techniques continue to be important as spacecraft constellations become both larger and more common and access to space becomes cheaper and more accessible. As a result, there are both more opportunities for spacecraft rendezvous missions, such as on-orbit spacecraft servicing or debris removal, and a greater need for spacecraft to be able to operate safely in close proximity to each other. These provide a useful case where the faster convergence of finite-time controllers can be beneficial. By rejecting perturbations faster than asymptotic controllers the overall safety of a spacecraft formation is increased.

For each scenario in this work, a spacecraft system is considered which consists of the spacecraft being controlled and a reference orbit. The spacecraft uses one of the control laws under consideration to maneuver to a designated state, generally the origin, relative to the reference orbit. This approximates a spacecraft rendezvous problem where the "chaser" spacecraft is controlled to rendezvous with a passive "target" spacecraft.

This approach can be expanded to include multiple spacecraft operating in formation by, for example, defining trajectories or waypoints for each individual spacecraft to follow using the finite-time control laws. While Chapter 4 does have the control being applied to follow a trajectory, the other control laws only target a single state.

1.4 Dissertation Outline

This dissertation is divided into the following chapters.

1.4.1 Background

In Chapters 2 and 3, the necessary background material for this dissertation is presented. Chapter 2 presents an overview of stability definitions and the Lyapunov conditions for stability. Additionally, a brief summary of some common linear and nonlinear control design approaches are introduced.

Each control law developed in this dissertation is applied to a spacecraft rendezvous problem for testing. Chapter 3 details the dynamics needed to understand this system from both an inertial perspective, as viewed from the Earth, and a relative perspective, where the motion of one spacecraft is viewed with respect to the other. Additionally, separation of the in-plane and cross-track dynamics and their transformation to chained-integrator systems through the Brunovsky transformation is detailed.

1.4.2 Tube-Based Model Predictive Control

In Chapter 4, a robust version of model predictive control (MPC), known as tube-based MPC [30], is developed for the spacecraft system. Standard MPC works by repeatedly solving an optimization problem at each control update step. The optimization problem generally considers the dynamics, state and control constraints, and an objective function that weights how heavily the MPC should seek to minimize state errors and control usage and finds the state and control trajectories that solve the problem. The first step of the control solution is then applied to the system and the process repeats itself. Tube-based MPC adds a linear feedback term to the standard MPC which, when properly selected, guarantees the system will satisfy the state and control constraints in the presence of perturbations.

For this work, the perturbations of the spacecraft rendezvous problem are estimated and then used to define a tube-based MPC which can keep the spacecraft within the allowable bounds. It is then tasked with following a guidance trajectory which calculates a correction maneuver over a fixed time horizon. This results in a system which has predefined-time stability for the nominal system and is able to asymptotically reject perturbations.

1.4.3 Common Lyapunov Control

Chapter 5 develops a common Lyapunov control [31] which finds a common Lyapunov function that stabilizes both the system of interest and a second reference system. This Lyapunov function can then be used to calculate a control which guarantees finite-time convergence. The connection between the common Lyapunov functions and the convergence time and control bounds are developed. Then, a semidefinite program is developed that finds a suitable set of Lyapunov functions while also minimizing the convergence time and control bounds.

1.4.4 Backstepping Control

A predefined-time backstepping control based on the work in [3] is presented in Chapter 6. This approach uses a tuning function that is selected to satisfy certain constraints at time $t = 0$ and $t = T_f$ where T_f is the predefined-time bound. Since any T_f can be chosen, and the tuning function selected to satisfy the conditions, the upper bound on the convergence for any initial condition is easily tunable.

Within these constraints, the tuning function can be selected to satisfy other requirements. This work examines the connection between the tuning function and the control magnitude applied to the system. Then, the tuning function is selected in such a way to minimize the maximum control magnitude while still satisfying the predefined-time condition.

1.4.5 Switching Surface Control

In Chapter 7, three different control laws based on switching surface principles are examined. In each case, the control law is dependent on the system's state relative to some set of surfaces and the applied control changes as a function of these surfaces.

The first control law is based on the minimum-time solution [32, 33] to the double integrator problem which results in the creation of a switching surface. The control law then applies the maximum allowable control magnitude with the direction determined by the system state, relative to the surface, and a single change in the direction when the surface is reached. This work applies this solution to the spacecraft rendezvous problem through the Brunovsky transformation. Additionally, the switching surface is optimized so that the maximum control is minimized over the full rendezvous trajectory.

The second and third control laws [2, 34] in this chapter are similar to each other in that they each drive the system towards, and keep it on, some surface which results in the convergence of the system to the origin. The switching function associated with this surface is, nominally, a discontinuous function but the practical benefits of using a

continuous function are shown. Additionally, the design space of allowable parameters and the associated impact on the convergence time and control usage is explored.

1.4.6 Conclusions and Future Work

The final chapter, Chapter 8, summarizes the conclusions made in each of the preceding chapters. Additionally, the possible avenues for future work which build on this dissertation are discussed.

Code and Data Availability

The code used in this work can be found on GitHub [35] and a version archived at the time of publication is available at [36]. The specific simulation data presented in the results for this work are available at [37].

CHAPTER 2

CONTROLS BACKGROUND

2.1 Stability Definitions

Consider a system of the form

$$\dot{x} = f(x) \tag{2.1}$$

where $f : D \to \mathbb{R}^n$ is the dynamics, $D \subset \mathbb{R}^n$ is an open neighborhood of the origin, and $f(0) = 0$. The origin is then referred to as an equilibrium point.

We introduce various notions of stability according to the following definitions. Each definition is assumed to only hold on some neighborhood of the equilibrium point resulting in local stability of the appropriate type. If the definition holds for all $x \in \mathbb{R}^n$ then the equilibrium point is globally stable.

Definition 2.1.1 ([38, Definition 2.1])**.** The equilibrium point of (2.1) is *stable* if for all $\varepsilon > 0$ there exists some $\delta > 0$ such that for all $t \geq t_0$,

$$\|x(t_0)\| < \delta \implies \|x(t)\| < \varepsilon$$

The equilibrium point is *unstable* if it is not stable.

Definition 2.1.2 ([38, Definition 2.1])**.** The equilibrium point of (2.1) is *asymptotically stable* if it is stable and δ can be chosen such that

$$\|x(t_0)\| < \delta \implies \lim_{t\to\infty} x(t) = 0.$$

Definition 2.1.3 ([38, Definition 4.5])**.** The equilibrium point of (2.1) is *exponentially stable* if there exist positive constants c, k, and λ such that for all $\|x(t_0)\| < c$,

$$\|x(t)\| \leq k\|x(t_0)\|e^{-\lambda(t-t_0)}$$

Exponential stability has the same principle limitation as asymptotic stability, namely that there is no guarantee that the system will reach the equilibrium point in finite time. However, note that the norm of x is bounded by $k\|x(t_0)\|e^{-\lambda(t-t_0)}$ so the maximum possible value of $\|x(t)\|$ can be calculated for $t \geq t_0$ which can be useful, even if the exact value is unknown.

Definition 2.1.4 ([34, Definition 1][39, Definition 1])**.** The equilibrium point of (2.1) is *finite-time stable* if it is asymptotically stable and $\exists \delta > 0$ such that $\forall t \geq T(x_0), \forall ||x(t_0)|| < \delta, x(t, x_0) = 0$ where T is the settling-time function, whose domain is the region of attraction [38, Section 8.2] and codomain is $\mathbb{R}_{\geq 0}$.

The settling-time function may or may not be known for a given system. However, it is important to note that the convergence time may be arbitrarily large and, ultimately, effectively slower than an equivalent system which "only" has an asymptotically stable equilibrium point.

Definition 2.1.5 ([34, Definition 2])**.** The equilibrium point of (2.1) is *fixed-time stable* if it is finite-time stable and the settling-time function $T(x_0)$ is bounded, i.e., $\exists T_{\max} > 0$ such that $T(x_0) \leq T_{\max}$ for all $||x_0|| < \delta$.

2.2 Lyapunov Functions

One common way to analyze stability is through the use of Lyapunov functions. The definitions are given for systems where the dynamics are Lipschitz continuous, although similar definitions can be given for systems with a discountinuous right hand side using Filipov solutions [40].

Definition 2.2.1 ([38, p. 116])**.** A continuously differentiable function $V : D \to \mathbb{R}$ is a *Lyapunov function* for (2.1) if

$$V(0) = 0 \text{ and } V > 0 \text{ on } D \setminus \{0\} \qquad \text{(Positive Definite)} \tag{2.2}$$

$$\dot{V} \leq 0 \text{ on } D. \tag{2.3}$$

A Lyapunov function can be thought of, figuratively in some cases and literally in others, as the energy of the system. Since the derivative is non-increasing, the "energy" of the system will never increase and the system states must remain bounded. The following theorem formalizes this analogy.

Theorem 1 ([38, Theorem 4.1])**.** The origin of (2.1) is stable if there exists a Lyapunov function.

Example 2.2.1. Consider the system defined as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 \end{aligned} \tag{2.4}$$

which has a single equilibrium point at $x = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$. Also, consider a candidate Lyapunov function defined as

$$V = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2. \tag{2.5}$$

Note that V is positive definite with a value of 0 at the equilibrium point. Taking the derivative with respect to time and substituting the dynamics gives

$$\dot{V} = x_1\dot{x}_1 + x_2\dot{x}_2 \tag{2.6}$$

$$= x_1x_2 - x_2x_1 \tag{2.7}$$

$$= 0 \tag{2.8}$$

Thus, given an initial state x_0, the system will never leave the level surface with value $V(x_0)$ and the equilibrium point is stable. $\triangle$

If a Lyapunov function with a more restrictive derivative constraint can be found then asymptotic stability is guaranteed.

Theorem 2 ([38, Theorem 4.1])**.** Let $V : D \to \mathbb{R}$ be a continuously differentiable function such that

$$V(0) = 0 \text{ and } V > 0 \text{ on } D \setminus \{0\} \qquad \text{(Positive Definite)} \tag{2.9}$$

$$\dot{V}(0) = 0 \text{ and } \dot{V} < 0 \text{ on } D \setminus \{0\} \qquad \text{(Negative Definite)} \tag{2.10}$$

Then, $x = 0$ is asymptotically stable.

Now, the system cannot remain at its present Lyapunov function value but must decrease over time, tending towards the $V(x) = 0$ state which is already defined to be the equilibrium point.

Example 2.2.2. Consider the system

$$\dot{x} = \begin{bmatrix} 0 & 3 \\ -2 & -1 \end{bmatrix} x \tag{2.11}$$

and a candidate Lyapunov equation

$$V = x^T \begin{bmatrix} \frac{11}{12} & \frac{1}{4} \\ \frac{1}{4} & \frac{5}{4} \end{bmatrix} x. \tag{2.12}$$

Since the eigenvalues of the symmetric matrix in the candidate Lyapunov function are positive, the function is positive definite which fulfills the first requirement. Taking the

time derivative of the candidate Lyapunov function and substituting in the system dynamics gives

$$\begin{aligned}
\dot{V} &= x^T \begin{bmatrix} \frac{11}{12} & \frac{1}{4} \\ \frac{1}{4} & \frac{5}{4} \end{bmatrix} \dot{x} + \dot{x}^T \begin{bmatrix} \frac{11}{12} & \frac{1}{4} \\ \frac{1}{4} & \frac{5}{4} \end{bmatrix} x \\
&= x^T \left(\begin{bmatrix} \frac{11}{12} & \frac{1}{4} \\ \frac{1}{4} & \frac{5}{4} \end{bmatrix} \begin{bmatrix} 0 & 3 \\ -2 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 3 \\ -2 & -1 \end{bmatrix}^T \begin{bmatrix} \frac{11}{12} & \frac{1}{4} \\ \frac{1}{4} & \frac{5}{4} \end{bmatrix} \right) x \\
&= x^T \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} x \\
&= -x_1^2 - x_2^2
\end{aligned} \tag{2.13}$$

which is negative for all $x_1, x_2 \neq 0$ and zero for $x_1 = x_2 = 0$. Thus, the derivative of the candidate Lyapunov function is negative definite. Since we have a positive definite function with a negative definite derivative, the origin of the system is asymptotically stable. △

Example 2.2.3. Consider the same system as the previous example

$$\dot{x} = \begin{bmatrix} 0 & 3 \\ -2 & -1 \end{bmatrix} x \tag{2.14}$$

which was shown to be asymptotically stable.

Now consider an alternate candidate Lyapunov function

$$V = \frac{1}{2} x_1^2 + \frac{1}{2} x_2^2 \tag{2.15}$$

which is positive definite.

Taking the derivative gives

$$\begin{aligned}\dot{V} &= x_1\dot{x}_1 + x_2\dot{x}_2 \\ &= 3x_1x_2 - 2x_1x_2 - x_2^2 \\ &= (x_1 - x_2)x_2\end{aligned} \tag{2.16}$$

Now note that $\dot{V}$ is positive when $(x_1 - x_2) > 0$ and $x_2 > 0$. Thus, $\dot{V}$ is not negative definite and the system cannot be defined to be either asymptotically stable or unstable based on this choice of V. △

As shown in Example 2.2.3, the fact that a system is asymptotically stable does not mean that any positive definite function will work as a Lyapunov function. Thus, a Lyapunov analysis can be used to prove that a system is stable but failure to find one cannot be used to prove the converse. Moreover, even if a system is stable, a suitable Lyapunov function may be non-trivial to discover.

This is likely the largest weakness of a Lyapunov-based approach for analyzing systems. While some classes of functions, such as quadratic functions, are generally effective Lyapunov candidates, there is no systematic approach to developing a Lyapunov function. Ultimately, a Lyapunov analysis comes down to the creativity and experience of the engineer and is very much both an art and a science.

Theorem 3 ([41, Lemma 1])**.** Consider the system described in (2.1) and suppose there is a continuously differentiable function $V : D \to \mathbb{R}$. Additionally, suppose there are real numbers $\alpha > 0$ and $0 < \lambda < 1$ such that V is positive definite on D and $\dot{V} + \alpha V^\lambda \leq 0$ on D. Then the equilibrium point is *finite-time* stable.

Example 2.2.4. Consider the system, modified from the one found in [41],

$$\dot{x} = (1 - N)x - \beta|x|^\eta \text{sign}(x) \tag{2.17}$$

with constant values $N > 1$, $\beta > 0$, and $-1 < \eta < 1$. Additionally, consider a candidate Lyapunov function

$$\begin{aligned} V &= x^2 \\ \dot{V} &= 2x\dot{x} \end{aligned} \tag{2.18}$$

The Lyapunov derivative of the system is then

$$\dot{V} = 2(1-N)x^2 - 2\beta|x|^{\eta}\text{sign}(x)x \tag{2.19}$$

Since $N > 1$, the term $2(1-N)x^2$ is always negative meaning that

$$\begin{aligned} \dot{V} &\leq -2\beta|x|^{\eta}\text{sign}(x)x \\ &= -2\beta|x|^{\eta+1} \\ &= -2\beta V^{(\eta+1)/2} \end{aligned} \tag{2.20}$$

With $\beta > 0$, and $-1 < \eta < 1$, this satisfies the requirements for finite-time stability.

△

In general, finding a suitable Lyapunov function and control law that can guarantee finite-time stability is not a well-defined process. As a result, direct application of Lyapunov functions is generally not the easiest approach to take when trying to design a controller. Instead, various methods have been developed which generate both a control law and a suitable Lyapunov function for specific classes of systems. These can then be applied to any suitable system to obtain a control without the need to create a new Lyapunov function for each system. To better understand the finite-time approaches in later chapters, standard techniques for asymptotic stability are now presented. They form the foundation for the finite-time techniques.

2.3 Control Techniques for Linear Systems

While Lyapunov-based control is broadly applicable, there are specific techniques that can be applied to linear systems. Consider a linear time-invariant (LTI) system of the form

$$\dot{x} = Ax + Bu \tag{2.21}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $A \in \mathbb{R}^{n\times n}$, and $B \in \mathbb{R}^{n\times m}$ where the objective is to design some feedback control law of the form $u = \phi(x)$ where $\phi : \mathbb{R}^n \to \mathbb{R}^m$.

2.3.1 Controllability

We first address the controllability of the system.

Definition 2.3.1 ([42, Definition 9.1][43, Definition 12.2])**.** The linear system (2.21) is called *controllable* on $[t_0,\ t_1]$ if given two times $t_0 < t_1$, then for any initial state $x(t_0) = x_0$ there exists an input signal u such that the corresponding solution of (2.21) satisfies $x(t_1) = 0$.

The question of controllability can be answered by constructing the controllability matrix as

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \tag{2.22}$$

and applying the following theorem.

Theorem 4 ([43, Theorem 12.1])**.** The LTI system (2.21) is controllable if and only if

$$\text{rank } \mathcal{C} = n. \tag{2.23}$$

If the system has been verified to be controllable, then work can begin on developing the control law. Various approaches have been developed to easily determine suitable control laws without needing to explicitly test different Lyapunov functions. These approaches are based on the following Theorem.

Theorem 5 ([43, Theorem 8.2])**.** Given a linear time invariant system of the form $\dot{x} = Ax$ with $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$, the following statements are equivalent:

1. The system is asymptotically stable
2. The system is exponentially stable
3. All the eigenvalues of A have strictly negative real parts
4. For every symmetric positive definite matrix Q, there exists a unique solution P to the following Lyapunov equation:

$$A^T P + PA = -Q \tag{2.24}$$

5. There exists a symmetric positive definite matrix P for which the following Lyapunov matrix inequality holds:

$$A^T P + PA < 0 \tag{2.25}$$

From the third property, an approach to developing a suitable control law can be found. Given the system (2.21) let $K \in \mathbb{R}^{n \times m}$ be a feedback gain matrix such $u = -Kx$ resulting in the system being defined to be

$$\dot{x} = (A - BK)x. \tag{2.26}$$

Defining the closed-loop system as $\tilde{A} = A - BK$, the closed-loop eigenvalues can be computed as

$$\det\left(\lambda I - \tilde{A}\right) = 0 \tag{2.27}$$

which will give a characteristic equation as a function of the elements of K and the eigenvalues. Provided the (A, B) system is controllable, the feedback terms can then be selected to match some nominal characteristic equation composed of the desired eigenvalues [43].

Example 2.3.1. Consider the system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{2.28}$$

where the desired pole placement for the closed-loop feedback system is $s = -1 \pm 1i$.

Introducing the feedback gain $K = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$ with the control law being $u = -Kx$ gives the closed-loop system as

$$\begin{aligned} \tilde{A} &= \begin{bmatrix} 0 & 1 \\ -1 & -3 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ -1-k_1 & -3-k_2 \end{bmatrix} \end{aligned} \tag{2.29}$$

The poles of the system are then found as the solution to

$$\begin{aligned} \det\left(sI - \tilde{A}\right) &= 0 \\ \det\left(\begin{bmatrix} s & -1 \\ 1+k_1 & s+3+k_2 \end{bmatrix}\right) &= 0 \\ s^2 + (3+k_2)s + (1+k_1) &= 0 \end{aligned} \tag{2.30}$$

Since the desired pole placement is known, we can find the desired characteristic equation as

$$\begin{aligned} (s+1+1i)(s+1-1i) &= 0 \\ s^2 + s - si + s + 1 - 1i + si + 1i + 1 &= 0 \\ s^2 + 2s + 2 &= 0 \end{aligned} \tag{2.31}$$

Comparing the two characteristic equations the necessary feedback gains to obtain the desired pole placement are found to be $k_1 = 1$ and $k_2 = -1$ with the closed-loop system

found to be

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ -1 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ u &= \begin{bmatrix} 1 & -1 \end{bmatrix} x \end{aligned} \tag{2.32}$$

△

2.3.2 Optimal Control

While setting the poles of a system, as shown in the preceding section, is an effective means of finding a stabilizing control law, there are generally additional factors that should be considered. These can include the desired convergence time, allowable control inputs, state deviations, and the balance among these factors. The optimal control framework gives a way to express these factors, as well as additional constraints, and find the control, if it exists, which satisfies them.

Optimal control can be applied to continuous or discrete systems which have linear or nonlinear dynamics. This section will present the problem for a continuous system with general dynamics function but not detail the corresponding methods for discrete systems. As will be seen in the next section, when using optimal control with linear systems, additional methods are available to find solutions.

A general problem is stated as follows [32]:

$$\min \quad J = \phi(t_f, x_f) + \int_{t_0}^{t_f} \ell(t, x, u)\, dt \tag{2.33}$$

$$\text{s.t.} \quad \dot{x} = f(t, x, u) \tag{2.34}$$

$$\psi(t_f, x_f) = 0 \tag{2.35}$$

$$x(0) = x_0 \tag{2.36}$$

$$u(t) \in \Omega \tag{2.37}$$

where $\phi : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the terminal cost, $\ell : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the running cost, $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ are the system dynamics, $\psi : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^p$ are the p terminal constraints, $x_0 \in \mathcal{X}_0 \subset \mathbb{R}^n$ is the initial system state, and $\Omega \in \mathcal{U} \subset \mathbb{R}^m$ is the allowable set of control inputs. The objective function is designed to account for the desired behavior of the system. By adding additional weight to the control portion of ℓ, for example, the resulting control law is incentivized to reduce the control usage although this will likely result in additional state deviations.

When solving an optimal control problem, two functions must first be introduced. First, the Hamiltonian, defined as

$$H(t, x, u, \lambda_0, \lambda) = \lambda_0 \ell(t, x, u) + \lambda^T f(t, x, u) \tag{2.38}$$

and second, the endpoint function, defined as

$$G(t_f, x_f, \lambda_0, \nu) = \lambda_0 \phi(t_f, x_f) + \nu^T \psi(t_f, x_f) \tag{2.39}$$

where $\lambda_0 \in \{0, 1\}$, λ is a continuous function with values in $\mathbb{R}^n$, and $\nu \in \mathbb{R}^p$ is a constant value with dimension equal to the number of terminal constraints.

From these two equations, the following necessary conditions for optimality can be derived.

$$\dot{\lambda} = -\frac{\partial H}{\partial x} \qquad \text{(Costate Equation)}$$

$$\dot{H} = \frac{\partial H}{\partial t} \qquad \text{(Hamiltonian Equation)}$$

$$\lambda_f = \frac{\partial G}{\partial x_f} \qquad \text{(Transversality Condition)}$$

$$H_f = -\frac{\partial G}{\partial t_f} \qquad \text{(Transversality Condition)}$$

$$u \in \text{argmin}_{\omega \in \Omega} H(t, x, \omega, \lambda_0, \lambda) \qquad \text{(Pointwise Minimum Condition)}$$

$$(\lambda_0, \lambda) \neq 0 \qquad \text{(Non-triviality Condition)}$$

A similar set of conditions exist for discrete systems.

Example 2.3.2 ([32, Example 5.2])**.** Consider the rectilinear motion dynamics

$$\dot{x} = u \tag{2.40}$$

with an initial position of $x(0) = 0$. The objective is to reach a desired final position of $x(t_f) = 1$ at a desired arrival time of $t_f = t_f^*$ while minimizing control usage.

The problem can be described as

$$\begin{aligned} \min \quad & \int_0^{t_f} u^2 \, dt \\ \text{s.t.} \quad & \dot{x} = u \\ & x(0) = 0 \\ & x(t_f) = 1 \\ & t_f = t_f^* \end{aligned} \tag{2.41}$$

Constructing the Hamiltonian and endpoint functions gives

$$\begin{aligned} H &= \lambda_0 u^2 + \lambda u \\ G &= \nu_1(x(t_f) - 1) + \nu_2(t_f - t_f^*) \end{aligned} \tag{2.42}$$

which results in the costate, Hamiltonian, and transverality conditions being

$$\begin{aligned} \dot{\lambda} &= 0 \\ \dot{H} &= 0 \\ \lambda_f &= \nu_1 \\ H_f &= -\nu_2 \end{aligned} \tag{2.43}$$

From these conditions, we can see that λ and H are both constant. Examining the Hamiltonian equation, we can see that if H and λ are constant (and λ_0 is always constant) then u must necessarily be constant as well.

Examining the pointwise minimum condition gives

$$u \in \text{argmin } \lambda_0 u^2 + \lambda u. \tag{2.44}$$

Using standard optimizaiton techniques, we can take the derivative with respect to u and find the zero giving

$$0 = \lambda_0 u + \lambda \tag{2.45}$$

If $\lambda_0 = 0$, then the only possible solution is $\lambda = 0$ which violates the triviality constraint. Thus, $\lambda_0 = 1$. Therefore, the minimum is found at $u = -\lambda$.

From the equation of motion, we get

$$\begin{aligned} x(t_f) &= \int_0^{t_f} u \, dt \\ &= u \, t_f \end{aligned} \tag{2.46}$$

which, when applying the final desired state and time, gives

$$u = \frac{1}{t_f^*} \tag{2.47}$$

Using this solution gives

$$\begin{aligned} \lambda &= -\frac{1}{t_f^*} \\ H &= -\frac{1}{2}\frac{1}{t_f^{*2}} \\ \nu_1 &= -\frac{1}{t_f^*} \\ \nu_2 &= \frac{1}{2}\frac{1}{t_f^{*2}} \end{aligned} \tag{2.48}$$

showing that all the conditions are met.

$\triangle$

2.3.3 Linear Quadratic Regulator

As a subset of optimal control, we can develop a Linear Quadratic Regulator (LQR) which is a feedback control law that will stabilize the system while minimizing an objective function consisting of state deviations and control usage terms. LQR is a very specific subset of the optimal control problem discussed above as it requires a linear system and is unable to include constraints. However, it does provide a way to balance control usage and state deviations while stabilizing the system.

We consider the continuous, linear optimal control problem where the system is of the form

$$\dot{x} = Ax + Bu \tag{2.49}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. We also introduce an objective function

$$J = \int_0^\infty x(t)^T Q x(t) + u(t)^T R u(t)\, dt \tag{2.50}$$

where $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are symmetric positive definite matrices. These represent the penalty associated with larger state error and control usage and allow the system response to be tuned between the size of the control input and the speed of the system response.

The control law can then be selected as shown in the following theorem, adapted from [43, Theorem 21.1] and [44, Section 9.2.3].

Theorem 6. Assume that $(A, B, Q^{1/2})$ is controllable and observable. Then, the solution to the optimal control problem can be expressed in state feedback form as

$$u = -Kx \tag{2.51}$$

with $K = R^{-1}B^T P$ where P is the symmetric, positive definite solution to the algebraic Riccati equation

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \tag{2.52}$$

This control stabilizes the system and minimizes the LQR objective function.

2.4 Control Techniques for Nonlinear Systems

Unlike linear systems, nonlinear systems are so varied in their structure that there is no general approach guaranteed to result in a suitable control law. However, there are some general techniques that can be attempted.

2.4.1 Feedback Linearization

Feedback linearization is a method where the nonlinearities in the system are canceled by the control law.

Example 2.4.1. Consider the system

$$\begin{aligned} \dot{x}_1 &= -x_1 - 3x_2 \\ \dot{x}_2 &= x_1 x_2^2 + u. \end{aligned} \tag{2.53}$$

Now, letting $u = v - x_1 x_2^2$, the system can be expressed as

$$\begin{aligned} \dot{x}_1 &= -x_1 - 3x_2 \\ \dot{x}_2 &= v \end{aligned} \tag{2.54}$$

which is a linear system and the control law for v can be developed using the linear control techniques discussed above. Letting $v = -x_2$, the linearized system is then

$$\begin{aligned} \dot{x}_1 &= -x_1 - 3x_2 \\ \dot{x}_2 &= -x_2 \end{aligned} \tag{2.55}$$

which is asymptotically stable.

Substituting v into u gives the control law for the actual, nonlinear system as

$$u = -x_2 - x_1 x_2^2. \tag{2.56}$$

△

The example shows a simple case, but it is not uncommon for systems to have nonlinearities in states which do not have direct control inputs. This precludes the approach used above where the nonlinearities are moved into the control but does not necessarily mean that feedback linearization is infeasible. As shown in the following definition, if an acceptable transformation can be found, then the system can be put into a form where feedback linearization is appropriate.

Definition 2.4.1 ([38, Definition 13.1])**.** A nonlinear system

$$\dot{x} = f(x) + g(x)u \tag{2.57}$$

where $f : D \to \mathbb{R}^n$ and $g : D \to \mathbb{R}^{n\times m}$ are sufficiently smooth on a domain $D \subset \mathbb{R}^n$ is said to be feedback linearizable if there exists a diffeomorphism $T : D \to \mathbb{R}^n$ such that $D_z = T(D)$ contains the origin and the change of variables $z = T(x)$ transforms the system into the form

$$\dot{z} = Az + B\gamma(x)[u - \alpha(x)] \tag{2.58}$$

where (A, B) is controllable and $\gamma(x)$ is nonsingular for all $x \in D$.

The existence of a suitable transformation may not be immediately obvious, but the necessary and sufficient conditions are given in [38] as well as the conditions used to develop the transformation. While these are beyond the scope of this material, the following example shows an application which uses a transformation to obtain a feedback linearized system.

Example 2.4.2 ([38, p. 507])**.** Consider the system

$$\begin{aligned} \dot{x}_1 &= a \sin x_2 \\ \dot{x}_2 &= -x_1^2 + u \end{aligned} \tag{2.59}$$

where the control cannot directly be used to cancel out the nonlinearities in the x_1 dynamics.

However, we now apply the transformation

$$\begin{aligned} z_1 &= x_1 \\ z_2 &= a \sin x_2 \end{aligned} \tag{2.60}$$

and take the derivatives, giving

$$\begin{aligned} \dot{z}_1 &= \dot{x}_1 \\ &= a \sin x_2 \\ &= z_2 \\ \dot{z}_2 &= a \cos(x_2) \dot{x}_2 \\ &= a \cos(x_2)(-x_1^2 + u) \end{aligned} \tag{2.61}$$

Note that now the system takes the desired form $\dot{z} = Az + B\gamma(x)(u - \alpha(x))$ with

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\ B &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \gamma(x) &= a \cos(x_2) \\ \alpha(x) &= x_1^2 \end{aligned} \tag{2.62}$$

The linearizing control can be chosen as

$$u = x_1^2 + v \frac{1}{a \cos x_2} \tag{2.63}$$

which results in

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= v \end{aligned} \tag{2.64}$$

where v can be chosen using linear control techniques.

Examining the inverse transform gives

$$\begin{aligned} x_1 &= z_1 \\ x_2 &= \sin^{-1}\left(\frac{z_2}{a}\right) \end{aligned} \tag{2.65}$$

which is well defined for $-a \leq z_2 \leq a$.

△

2.4.2 Backstepping

Backstepping is a control method where the dynamics of the first state is individually considered as if it is directly controllable. The next state is then designed so that it satisfies the stability requirement for the previous state and is, itself, stable. This process continues until the final state, where the actual control enters the system, is reached.

Theorem 7 ([38, Lemma 14.2])**.** Consider the system

$$\dot{\eta} = f(\eta) + g(\eta)\xi \tag{2.66}$$

$$\dot{\xi} = u. \tag{2.67}$$

Let $\phi(\eta)$ be a stabilizing state feedback law for (2.66) with $\phi(0) = 0$ and $V(\eta)$ be a Lyapunov function that, for all $\eta \in D$, satisfies

$$\frac{\partial V}{\partial \eta}[f(\eta) + g(\eta)\phi(\eta)] \leq -W(\eta) \tag{2.68}$$

for some positive definite $W(\eta)$. Then, the state feedback control law

$$u = \frac{\partial \phi}{\partial \eta}[f(\eta) + g(\eta)\xi] - \frac{\partial V}{\partial \eta}g(\eta) - k[\xi - \phi(\eta)] \tag{2.69}$$

stabilizes the origin of the system with $V(\eta) + [\xi - \phi(\eta)]^2/2$ as a Lyapunov function.

We begin with a somewhat trivial example for clarity and then move to a more meaningful example.

Example 2.4.3. Consider the integrator system

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned} \tag{2.70}$$

We begin by considering the equation $\dot{x}_1 = x_2$ and treating x_2 as the input. Letting $x_2 = \phi(x_1)$ be the control law that stabilizes the $\dot{x}_1$ system we can choose $\phi(x_1) = -x_1$ with $V(x_1) = x_1^2/2$. This results in $\dot{V} = -x_1^2$ showing that it is stable.

We now introduce a change of variables, $z_2 = x_2 - \phi(x_1) = x_2 + x_1$ resulting in the system

$$\begin{aligned} \dot{x}_1 &= -x_1 + z_2 \\ \dot{z}_2 &= u - x_1 + z_2 \end{aligned} \tag{2.71}$$

with a combined Lyapunov function of

$$\begin{aligned} V_c &= \frac{1}{2}x_1^2 + \frac{1}{2}z_2^2 \\ \dot{V}_c &= -x_1^2 + z_2 u + z_2^2 \end{aligned} \tag{2.72}$$

By choosing $u = -2z_2$, the Lyapunov stability condition is satisfied.

Undoing the coordinate transformation shows the closed-loop system in the original coordinates to be

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 - x_2 \end{aligned} \tag{2.73}$$

The eigenvalues of this system are $-1 \pm i$ showing that the origin is globally exponentially stable.

$\triangle$

A second, less trivial, example is now shown to demonstrate how backstepping can be used to address nonlinearities that are not directly connected with the control input.

Example 2.4.4 ([38, Example 14.8])**.** Consider the system

$$\begin{aligned} \dot{x}_1 &= x_1^2 - x_1^3 + x_2 \\ \dot{x}_2 &= u \end{aligned} \tag{2.74}$$

Now, we examine only the first state of the system

$$\dot{x}_1 = x_1^2 - x_1^3 + x_2 \tag{2.75}$$

and treat x_2 as the control input. Let us choose $x_2 = -x_1^2 - x_1$ which gives the resulting dynamics as

$$\dot{x}_1 = -x_1^3 - x_1 \tag{2.76}$$

which is asymptotically stable as can be seen by using the Lyapunov function $V = \frac{1}{2}x_1^2$ giving

$$\dot{V} = -x_1^4 - x_1^2 \tag{2.77}$$

which is clearly negative definite in x_1.

We will define $\phi : \mathbb{R} \to \mathbb{R}$ to be $\phi(x_1) = x_2$ or

$$\phi = -x_1^2 - x_1 \tag{2.78}$$

We now backstep by introducing a new variable z defined as

$$\begin{aligned} z &= x_2 - \phi(x_1) \\ z &= x_2 + x_1^2 + x_1 \\ x_2 &= z - x_1^2 - x_1 \end{aligned} \tag{2.79}$$

Using the change of variables gives

$$\begin{aligned} \dot{x}_1 &= -x_1 - x_1^3 + z \\ \dot{z} &= \dot{x}_2 + 2x_1\dot{x}_1 + \dot{x}_1 \\ &= u + (2x_1 + 1)(-x_1 - x_1^3 + z) \end{aligned} \tag{2.80}$$

Using $V_c = \frac{1}{2}x_1^2 + \frac{1}{2}z^2$ gives the derivative as

$$\begin{aligned} \dot{V}_c &= x_1(-x_1 - x_1^3 + z) + z(u + (2x_1 + 1)(-x_1 - x_1^3 + z)) \\ &= -x_1^2 - x_1^4 + z(u + (2x_1 + 1)(-x_1 - x_1^3 + z) + x_1) \end{aligned} \tag{2.81}$$

Taking

$$u = -(2x_1 + 1)(-x_1 - x_1^3 + z) - x_1 - z \tag{2.82}$$

results in the Lyapunov function derivative being

$$\dot{V}_c = -x_1^2 - x_1^4 - z^2 \tag{2.83}$$

which is negative definite and thus the system is asymptotically stable.

Comparing the theorem to this example shows the following. For the original system, we have $x_1 = \eta$ and $x_2 = \xi$ with $f(x_1) = x_1^2 - x_1^3$ and $g(x_1) = 1$. The feedback law for $\dot{x}_1$, assuming x_2 is the control input, was defined to be $\phi(x_1) = -x_1^2 - x_1$ and was shown to be stabilizing using the Lyapunov function $V = \frac{1}{2}x_1^2$. Recall that $\dot{V} = -x_1^2 - x_1^4$ so choosing $W = x_1^2 + x_1^4$ gives a positive definite W such that for all $x_1 \in D$,

$$\begin{aligned} \dot{V} &= \frac{\partial V}{\partial x_1}[f(x_1) + g(x_1)\phi(x_1)] \\ &= -W(x_1) \end{aligned} \tag{2.84}$$

Examining the control law as stated in the theorem gives

$$\begin{aligned} u &= \frac{\partial \phi}{\partial x_1}[f(x_1) + g(x_1)x_2] - \frac{\partial V}{\partial x_1}g(x_1) - k[x_2 - \phi(x_1)] \\ &= (-2x_1 - 1)(x_1^2 - x_1^3 + x_2) - x_1(1) - k(x_2 + x_1^2 + x_1) \\ &= (-2x_1 - 1)(-x_1^3 + z - x_1) - x_1 - kz \\ &= -(2x_1 + 1)(-x_1^3 - x_1 + z) - x_1 - kz \end{aligned} \tag{2.85}$$

which matches the control law found previously with $k = 1$.

△

While the example only considers a two state system, backstepping can be applied to systems with multiple states assuming that they meet the requisite structure. Though not a complete survey of linear and nonlinear control techniques, they form the foundation of techniques in later chapters for finite-time control.

CHAPTER 3
SPACECRAFT DYNAMICS BACKGROUND

This work applies various control laws and approaches to rendezvous and proximity operations (RPO) applications which involve spacecraft operating in close proximity to each other. RPO applications arise in applications such as distributed spacecraft sensing, inspection, construction, refuelling, and docking. This chapter gives a brief overview of the necessary background required to understand the specific application of spacecraft RPO operations used.

In this chapter, we first describe the dynamics and perturbations associated with orbital motion. Then, the Clohessy-Wiltshire (CW) relative orbital dynamics are introduced. Finally, the Brunovsky transformation is introduced as a way to convert the relative orbital dynamics into a form that can easily be used with various control laws.

3.1 Two-Body Motion

The motion of a single spacecraft in orbit around the Earth is often described in the Earth Centered Inertial (ECI) frame. This frame is centered on the Earth with the X-axis pointing in the vernal equinox direction, the Z-axis aligned with the Earth's rotation, and the Y-axis completing a right-handed coordinate frame [45]. While the frame translates with the motion of the Earth, it does not rotate.

In this frame, the motion of a spacecraft can be described using the two-body model as

$$\begin{aligned} \dot{r} &= v \\ \dot{v} &= -\frac{\mu}{\|r\|^3} r + a_D + a_T \end{aligned} \tag{3.1}$$

where μ is the gravitational parameter of the two-body system, a_D is the disturbance acceleration, and a_T is the thrust acceleration. For the Earth, $\mu = 3.986\,004\,418 \times 10^{14}\,\mathrm{m}^3/\mathrm{s}^2$ [45].

The first term is the two-body term. It is the gravitational acceleration experienced by the spacecraft due to the Earth. While this is a primary effect on the spacecraft, a number of additional effects are also present to lessening degrees. These include gravity perturbations due to the central body not being a sphere, drag effects due to the body's atmosphere, solar radiation pressure due to the Sun, and third-body perturbations resulting from the gravitational influence of other bodies, such as the Sun or Jupiter, acting on the spacecraft [46]. Since the magnitude of these forces and accelerations vary, it is generally effective to only model the most impactful ones and assume the others are sufficiently insignificant with the cutoff being determined by the desired application.

In this work, a number of control laws are applied to RPO problems with the spacecraft assumed to be in a Low-Earth orbit (LEO). While a precise cutoff limit for what defines LEO is nebulous, it roughly encapsulates orbits with an altitude below 2000 km [47]. For this work, the orbits are well within that bound and exist around 500 km where the primary perturbations acting on a spacecraft are J2 gravity perturbations and drag.

3.1.1 Gravity Perturbations

Since the Earth is not sphere shaped, the gravitational force experienced by a spacecraft is not identical at every point along its orbit. Through experimentation, the gravitational field of the Earth has been fitted to a spherical harmonic model which gives the gravitational force expected depending on a spacecraft's position [48, 49]. Depending on the desired level of precision, this model can include as many terms as desired. The dominating coefficient in the model, J2, results from the Earth bulging slightly at the equator and being flatter at the poles and is what is used in this work.

In the ECI frame, the additional acceleration due to the J2 perturbation is [45]

$$a_{J2} = \frac{3}{2}\frac{J_2 \mu R_E^2}{\|r\|^4}\left[\frac{x}{\|r\|}\left(5\frac{z^2}{\|r\|^2}-1\right)\hat{i} + \frac{y}{\|r\|}\left(5\frac{z^2}{\|r\|^2}-1\right)\hat{j} + \frac{z}{\|r\|}\left(5\frac{z^2}{\|r\|^2}-3\right)\hat{k}\right] \quad (3.2)$$

where R_E is the radius of the Earth; J_2 is the J2 coefficient; $\hat{i}$, $\hat{j}$, and $\hat{k}$ are unit vectors pointing along each of the ECI axes; and, $r = \begin{bmatrix} x & y & z \end{bmatrix}^T$ is the position vector in ECI

space. This acceleration is then added to the nominal acceleration resulting from the nominal two-body motion.

3.1.2 Drag Perturbation

Although the Earth's atmosphere thins and becomes less dense as one's altitude increases, atmospheric drag is still a main cause of perturbations for spacecraft operating in the lower range of LEO. The drag model is given as [45]

$$a_{drag} = -\frac{1}{2}\rho\|v_{rel}\|\frac{C_D A}{m}v_{rel} \tag{3.3}$$

where ρ is the atmospheric density, v_{rel} is the velocity of the spacecraft relative to the Earth's atmosphere, C_D is the coefficient of drag, and A is the spacecraft's surface area. Although the model is conceptually straightforward, a number of simplifying assumptions are made.

The atmospheric density model uses a linear interpolation of the US Standard Atmosphere 1976 model and does not account for any effects due to solar activity [50]. The relative velocity of the spacecraft to the atmosphere is calculated as

$$v_{rel} = v - \omega_E \times r \tag{3.4}$$

where r and v are the spacecraft position and velocity in the ECI frame and ω_E is the angular velocity of the Earth. Furthermore, it is assumed that the atmosphere rotates perfectly with the Earth.

The coefficient of drag C_D and surface area A are both highly dependent on the geometry and attitude of the spacecraft. Instead, a simplified "cannonball" model is used to approximate the spacecraft as a sphere which then has a surface area and C_D independent of the actual orientation.

Once the drag perturbation is calculated, it is added to the nominal accelerations acting on the spacecraft.

3.1.3 Thruster Dynamics

For this work, it is assumed that the spacecraft has individual thrusters acting along each axis of the spacecraft. Additionally, no attitude dynamics are modeled and it is assumed that the spacecraft can instantaneously orient itself to achieve the desired thrust vectors.

Given a desired acceleration of $a_d \in \mathbb{R}^3$, and the spacecraft's current mass m, the necessary thrust $F_T \in \mathbb{R}^3$ is

$$F_T = ma_d \tag{3.5}$$

Then, the change in mass can be modeled as [45, Chapter 11],

$$\dot{m} = \frac{1}{g_0 I_{sp}} \|F_T\|_1 \tag{3.6}$$

where $g_0 = 9.81\,\mathrm{m/s}$ is the standard gravity and I_{sp} is the specific impulse of the engine. Note that the 1-norm is used to calculate the total thrust usage since we are assuming thrust along three orthogonal axes.

With the mass changing over time, we can now calculate the total ΔV used over the course of a simulation. ΔV is a measure of the total control used by a spacecraft which accounts for engine performance and the mass of the spacecraft which enables it to be a standard point of comparison.

Given the initial mass of a spacecraft m_0 and the final mass m_f, the ΔV is calculated as ([45], Chapter 11)

$$\Delta V = I_{sp} g_0 \ln \frac{m_0}{m_f} \tag{3.7}$$

Table 3.1: Spacecraft physical parameters used for simulating the dynamics.

Parameter	Value
Initial Mass	24 kg
Surface Area	0.048 m^2
C_D	2.2
I_{sp}	160 s

3.1.4 Spacecraft Parameters

One common spacecraft form factor is the CubeSat design [51]. These are defined in terms of a base unit, known as a 'U', which is a cube with edge lengths of 10 cm. From this, larger spacecraft designs are defined based on how many base units it is comprised of, such as 2U, 6U, and 12U .

The standardized approach for these small satellites gives rise to numerous benefits which have increased access to space [52]. CubeSats are relatively cheap to build when compared with traditional spacecraft due to their small size and use of commercially available parts. The standardization of the form factor means that multiple CubeSats can be deployed from a single launch vehicle which defrays the launch cost for any single CubeSat. Furthermore, due to the small form factor, CubeSats can be launched as secondary payloads to larger, traditional satellites.

For this work, the physical spacecraft parameters are based on a 12U CubeSat design [51] and shown in Table 3.1. Additionally, a propulsion system is assumed based on a hybrid propulsion system designed for CubeSats [53] with an I_{sp} of 160 s.

3.2 Relative Spacecraft Motion

While computing the motion of a spacecraft in the ECI is useful, in RPO applications it is generally more helpful to understand the relative motion of one spacecraft to another. To this end, a linearized model known as the Clohessy-Wiltshire (CW) equations were developed.

3.2.1 Local Vertical, Local Horizontal Frame

The CW equations are developed in the relative Local-Vertical, Local-Horizontal (LVLH) frame. The LVLH frame is centered on a spacecraft in orbit with the X-axis pointing along the spacecraft's ECI position vector, the Z-axis pointing along the spacecraft's orbit's angular momentum vector, and the Y-axis completing a right-hand rule. Given r and v of the reference spacecraft in ECI space, this is calculated as [45]

$$\begin{aligned} \hat{i}_{LVLH} &= \frac{r}{\|r\|} \\ \hat{k}_{LVLH} &= \frac{r \times v}{\|r \times v\|} \\ \hat{j}_{LVLH} &= \hat{k}_{LVLH} \times \hat{i}_{LVLH} \end{aligned} \tag{3.8}$$

with the transformation matrix from ECI to LVLH being

$$T = \begin{bmatrix} \hat{i}^T_{LVLH} \\ \hat{j}^T_{LVLH} \\ \hat{k}^T_{LVLH} \end{bmatrix} \tag{3.9}$$

Given two spacecraft with ECI position states r_A, r_B, velocity states v_A, v_B, and acceleration states a_A, a_B, the relative states are calculated in the ECI frame as [45]

$$\begin{aligned} \Omega &= \frac{r_A \times v_A}{\|r_A\|^2} \\ \dot{\Omega} &= -2\frac{v_A \dot{r}_A}{\|r_A\|^2}\Omega \\ r_{rel} &= r_B - r_A \\ v_{rel} &= v_B - v_A - \Omega \times r_{rel} \\ a_{rel} &= a_B - a_A - \dot{\Omega} \times r_{rel} - \Omega \times (\Omega \times r_{rel}) - 2\Omega \times v_{rel} \end{aligned} \tag{3.10}$$

This can then be converted to the LVLH frame as

$$\begin{aligned} r_{LVLH} &= T r_{rel} \\ v_{LVLH} &= T v_{rel} \\ a_{LVLH} &= T a_{rel} \end{aligned} \tag{3.11}$$

3.2.2 Clohessy-Wiltshire Equations

The CW dynamics describe the motion of one spacecraft relative to a second in an LVLH frame centered on the reference spacecraft. Since the CW dynamics are a linear approximation of the orbital motion, their accuracy degrades as the distance between the two spacecraft increases. Additionally, the linearization assumes a circular orbit so accuracy degrades as the eccentricity of the reference orbit increases. For all the applications used in this work the CW dynamics provide satisfactory levels of accuracy and are used for the control dynamics.

The dynamics themselves are expressed as [45, 49]

$$\begin{aligned} \dot{x} &= Ax + Bu \\ A &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\ 0 & 0 & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \end{bmatrix} \\ B &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3.12}$$

where ω is the mean orbital motion of the spacecraft defined to be

$$\omega = \sqrt{\frac{\mu}{a^3}} \tag{3.13}$$

with a being the semi-major axis of the reference orbit.

The X- and Y-axes are often referred to as the in-plane dynamics, while the Z-axis is referred to as the out-of-plane, or cross-track, dynamics. Note that the in-plane and out-of-plane dynamics are decoupled and can be controlled separately. Examining the in-plane dynamics, the system is completely controllable with controls along the X- and Y-axes. Similarly, the out-of-plane dynamics are also completely controllable with control along the Z-axis.

More interestingly, examining the controllability matrix for the in-plane dynamics with control only along the Y-axis gives

$$\begin{aligned} C &= \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 2\omega & 0 \\ 0 & 1 & 0 & -4\omega^2 \\ 0 & 2\omega & 0 & -2\omega^3 \\ 1 & 0 & -4\omega^2 & 0 \end{bmatrix} \end{aligned} \tag{3.14}$$

with $\text{rank}(C) = 4$ showing that the in-plane dynamics are completely controllable using only control along the Y-axis. Since the force due to drag acts along the Y-axis, this means that differential drag techniques can be used to control the in-plane dynamics without the use of any propellant.

The case where both X- and Y-axes have allowable control inputs will be referred to as the dual-input, in-plane dynamics. Similarly, the case where control inputs are only allowed along the Y-axis will be referred to as the single-input, in-plane dynamics. For the out-of-plane dynamics, control is always assumed to enter along the Z-axis.

3.3 CW Brunovsky Transformation

Given a linear system, there are multiple ways to describe the system in a form

$$\dot{x} = Ax + Bu \tag{3.15}$$

where $x \in \mathbb{R}^n$ is the states and $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ and the dynamics and control matrices, respectively [43]. However, depending on the situation, some realizations are more useful than others. Some common realizations include minimum realizations, controllable canonical form, and observable canonical form.

One form that is used repeatedly in this work is the Brunovsky transformation which converts a system into a chain of integrators with the control appearing on the last state [31]. This results in a simple system allowing for easy application of control laws.

Writing the original system as

$$\dot{x} = Ax + Bu \tag{3.16}$$

with $x \in \mathbb{R}^n$ and $u \in \mathbb{R}$, the Brunovsky form can be found as

$$\begin{aligned} A_B &= T_B^{-1}(A + BC)T_B \\ B_B &= T_B^{-1}B \\ z &= T_B^{-1}x \\ v &= u - Cx \\ \dot{z} &= A_B z + B_B v \end{aligned} \tag{3.17}$$

where C is the Brunovsky linear feedback term, T_B is the transform matrix, u is the control of the original system, and v is the control for the Brunovsky form. If C and T_B are chosen

properly then A_B and B_B will reduce to the Brunovsky forms

$$A_B = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \tag{3.18}$$

$$B_B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.19}$$

While this is for a single input system, multi-input systems can also be transformed. The resulting system then consists of independent subsystems which each take the form

$$\dot{z} = A_B z + B_B v \tag{3.20}$$

with the dimensions varying depending on the underlying system. The control v can then be determined with a suitable control law which acts on the Brunovsky system A_B and B_B using the transformed state $z = T_B^{-1} x$. The control u, which is actually applied to the system, is then found as $u = v + Cx$. In [54], a method is given for how to compute the Brunovsky form but the CW system is simple enough that that method was not used in this work.

For the out-of-plane dynamics, the Brunovsky form is obtained by moving the $n^2 z$ term into the control resulting in

$$\begin{aligned} T_{B,z} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ C_z &= \begin{bmatrix} \omega^2 & 0 \end{bmatrix} \end{aligned} \tag{3.21}$$

For the in-plane dynamics, there are two possible Brunovsky transformations depending on how many control inputs are available. If controls are allowed along both the X- and Y-axes, the transformation is found to be

$$T_{B,xy} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad C_{xy} = \begin{bmatrix} -3\omega^2 & 0 & 0 & -2\omega \\ 0 & 0 & 2\omega & 0 \end{bmatrix} \tag{3.22}$$

which produces the Brunovsky form

$$\dot{x}_{xy} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} x_{xy} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} v_{xy} \tag{3.23}$$

Note that this creates two independent subsystems, and separate controllers can be designed to find the Brunovsky controls for each subsystem. However, as a result of the transformation, the actual controls applied to the in-plane dynamics are not independent of each other.

If we consider the case where only the Y-axis is used to control the in-plane dynamics, then the Brunovsky form is obtained with

$$T_{B,y} = \begin{bmatrix} 0 & 2\omega & 0 & 0 \\ -3\omega^2 & 0 & 1 & 0 \\ 0 & 0 & 2\omega & 0 \\ 0 & -3\omega^2 & 0 & 1 \end{bmatrix} \qquad C_y = \begin{bmatrix} 0 & 0 & \frac{\omega}{2} & 0 \end{bmatrix} \tag{3.24}$$

Once a suitable Brunovsky transformation is found, a control law can be designed using the simplified form and used to find the control v. This can then be converted using Equation 3.17 to find the control u which is then applied to the nominal system. While the Brunovsky form allows for easy implementation of control laws due to the simple dynamics, the use of the feedback gain C in the transformation does introduce a complicating factor.

Since this is determined when performing the Brunovsky transformation, it exists outside any control law design and its effects on the system response cannot be modified. While this is not necessarily a problem, it does introduce difficulties when attempting to shape the control inputs to achieve some goal such as minimizing or constraining control.

3.4 Simulation Implementation

In this work, the simulations are propagated using either using a fixed step Runge-Kutta 4^{th} order solver or an adaptive step 4^{th} and 5^{th} order solver. The control laws are implemented using a sample-and-hold approach where the states at each step are measured, the control input is calculated, and then the control is held constant over the integration timestep. This approach to integration and control updates requires the proper selection of timesteps. If the integration timesteps are too large, then higher frequency effects in the system will not be accurately represented and the propagation will diverge from the true dynamics. Similarly, if the control update timesteps are too long, the system will be unable to respond to perturbations. Conversely, as the timesteps decrease, the computational resources required to simulate the controlled system increase. This can result in problems simulating the system within a reasonable time frame and may not accurately reflect the response rate of physical actuators. This section provides a brief analysis to justify the selection of the timesteps used in this work.

Note that while the Runge-Kutta solvers sample the dynamics at subincrements of the integration timestep, the control input is held constant over the full timestep and is only based on the initial states at the start of the integration step. The sample-and-hold approach calculates and applies the control inputs over the full integration step and does not change, even when an adaptive step solver is used. Thus, the adaptive step can help improve the

accuracy in the integration of the dynamics, but it does not change the system performance by increasing the update rate of the control law. Again, the choice between fixed or adaptive step integrators is made with the goal of balancing accuracy and computational requirements.

An analysis is first performed to determine how increasing the integration timestep of the simulation increases the error. The perturbed, uncontrolled two-body dynamics are simulated over 1.5 hrs, roughly a single orbit, with each simulation using a timestep that ranges from 1×10^{-3} s to 10 s. The dynamics are propagated using a fixed step Runge-Kutta 4^{th} order solver with the initial conditions for each simulation shown in Table 3.2 and spacecraft properties as presented in Table 3.1. Each trajectory is sampled at a 10 s rate, corresponding to the longest timestep, and compared to the trajectory with a 1×10^{-3} s timestep. This simulation with the 1×10^{-3} s timestep is used as the reference and represents the true trajectory that would be expected for the actual system.

Table 3.2: Orbital elements used to define the reference orbit for the timestep analysis.

Parameter	Value
Semimajor Axis	6878 km
Eccentricity	1×10^{-4}
Inclination	25°
Right Ascension of the Ascending Node	45°
Argument of Perigee	0°
True Anomaly	0°

The difference between each trajectory and the reference are shown in Figure 3.1. For each simulation, the error increases over time relative to the 1×10^{-3} s reference, resulting from the accumulation of small errors in the integration. The simulations with timesteps of 1 s or less are generally indistinguishable and have a final position error on the order of 10^{-5} m. For the timesteps of 5 s and 10 s the errors are noticeably larger, with the 10 s timestep having a final error on the order of 10^{-2} m. This indicates that these larger timesteps are not able to fully capture the perturbations, although for this work, an integration error of 10^{-2} m over the course of an orbit is acceptable. Overall this shows that the integrated dynamics are sufficiently close to the actual dynamics.

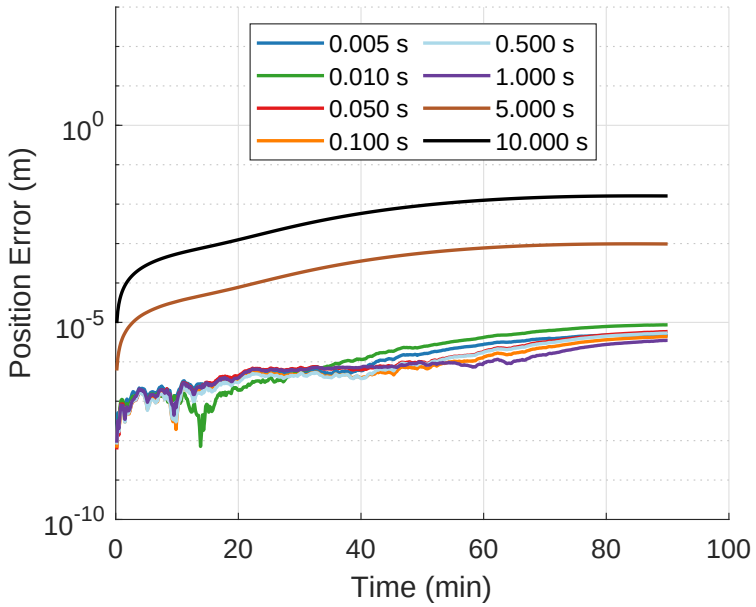

Fig. 3.1: Position errors for the perturbed two-body dynamics over a range of timesteps. Errors are calculated relative to a simulation with a timestep of 1×10^{-3} s.

In order to explore the effect of changing the control update timestep, two different controllers are tested with timesteps ranging from 1×10^{-3} s to 10 s. As with the integration error analysis, the resulting trajectories are sampled every 10 s and then compared to the simulation with the 1×10^{-3} s step. Each control is implemented as a sample-and-hold, where the control input is calculated at each timestep and then held constant until the next update. In each simulation, the integration step and control update have identical timesteps.

The first control law is a linear feedback control designed around the continuous CW dynamics of the form

$$\dot{x} = Ax + Bu \tag{3.25}$$

$$u = -Kx \tag{3.26}$$

where $A \in \mathbb{R}^{6\times6}$ and $B \in \mathbb{R}^{6\times3}$ correspond to the CW matrices and $K \in \mathbb{R}^{3\times6}$ is a feedback matrix. For this analysis, the K matrix is selected such that the eigenvalues of the closed

loop system are

$$\lambda(A - BK) = -\begin{bmatrix} .06 & .05 & .04 & .03 & .02 & .01 \end{bmatrix} \tag{3.27}$$

which shows the closed-loop system is stable since all the eigenvalues have strictly negative real parts [43, Theorem 8.2]. Note that although the control law is designed assuming a continuous system, the control is only updated at discrete times.

A second control law was designed around the discretized CW dynamics

$$x[k+1] = A_D x[k] + B_D u[k] \tag{3.28}$$

where $A_D \in \mathbb{R}^{6\times 6}$ and $B_D \in \mathbb{R}^{6\times 3}$ are found as [43, Section 6.1]

$$A_D = \exp\left(A\Delta t\right) \tag{3.29}$$

$$B_D = \int_0^{\Delta t} \exp\left((\Delta t - \tau)A\right) B \, d\tau \tag{3.30}$$

with δt being the control update timestep. Note that this approach assumes the control $u[k]$ is constant over the period from step k to $k+1$.

Similar to the continuous dynamics, a feedback control law can be developed as

$$u[k] = -K_D x[k] \tag{3.31}$$

with the requirement that the eigenvalues of the closed loop system exist within the unit circle [43, Theorem 8.4]. The feedback matrix K_D is then selected so the eigenvalues of the closed-loop system are

$$\lambda(A_D - B_D K_D) = -\begin{bmatrix} .06 & .05 & .04 & .03 & .02 & .01 \end{bmatrix} \tag{3.32}$$

which satisfies the stability requirement. This control law approach is tested over the same range of timesteps as the continuous control. Since A_D and B_D are dependent on the timestep selection, the control law is recalculated for each choice of timestep.

The continuous and discretized linear feedback controllers are both tested using the perturbed two-body dynamics. For each simulation, a desired inertial orbit is defined as shown in Table 3.2 with the initial states of the spacecraft set as

$$x_0 = \begin{bmatrix} 390\,\text{m} & 390\,\text{m} & 390\,\text{m} & 1.5\,\text{m/s} & 1.5\,\text{m/s} & 1.5\,\text{m/s} \end{bmatrix}^T \tag{3.33}$$

relative to the desired orbit in the LVLH frame. At each control update step, the LVLH state of the spacecraft relative to the desired orbit is found and used to calculate the feedback control. This is then applied to the perturbed two-body dynamics.

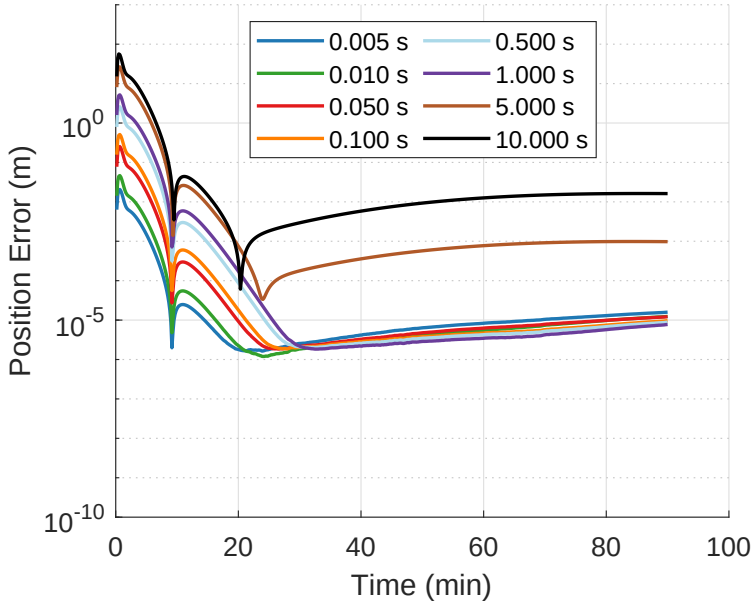

Fig. 3.2: Position error relative to the reference trajectory with a continuous time feedback control law. The reference trajectory is updated with a 0.001 s stepsize.

The results for the continuous control are presented in Figure 3.2 and show that each of the simulations exhibit two distinct phases. Due to the differing timesteps, each simulation exhibits some initial error which decreases over time as the control law is able to drive the system to the desired trajectory. Note that the trajectories for each simulation are separated based on the control update timesteps, with the longer timesteps having higher

errors. Eventually each simulation reaches a point of minimum error between the 20 min and 30 min times, and then exhibits steady state behavior where the error only gradually rises over time. Comparing this steady-state behavior to the uncontrolled simulations (Figure 3.1) shows that the controlled errors are of the same order of magnitude as the uncontrolled errors. This suggests that after 30 min, the differences are driven by the integration error with little relative difference due to the controller timestep selection.

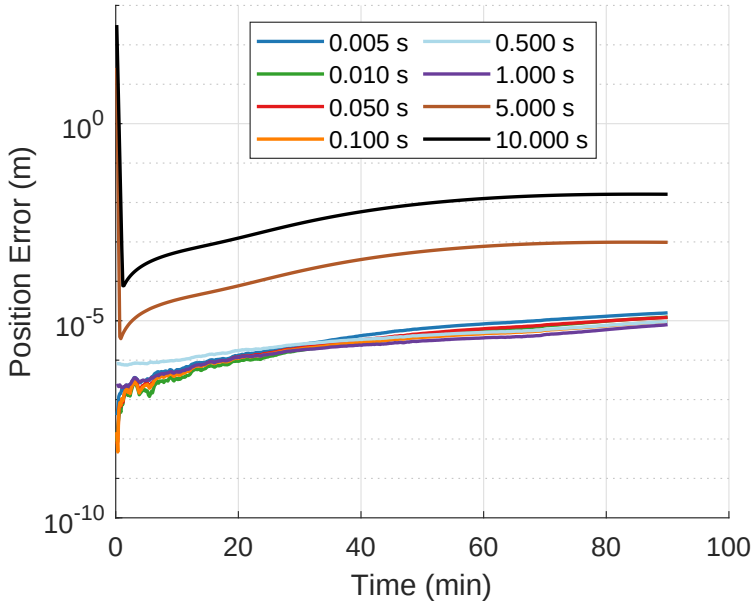

Fig. 3.3: Position error relative to the reference trajectory with a discrete time feedback control law. The reference trajectory is updated with a 0.001 s timestep.

The results for the discretized feedback control law are shown in Figure 3.3. In contrast to the continuous control law, there is much less separation between the different simulations with only the 5 s and 10 s timesteps having noticeable separation. Additionally, the simulations with timesteps less than 5 s do not show any transitory period where the trajectory differs from the reference. The 5 s and 10 s timestep simulations do initially diverge from the reference, but are able to converge much faster with the steady-state response being reached by about 1 min, rather than the 20 min that are seen in the continuous control. Also, the

steady-state response mirrors the uncontrolled results and suggests that the differences are driven by integration error, not controller performance.

Overall, the use of the discrete solution to the CW dynamics allows for better consistency across a range of timesteps since the control model is able to explicitly account for the impact of the control inputs. Thus, as the timesteps increase, there is not a significant jump in the error like there is with the continuous controller. The obvious exception to this is with the 5 s and 10 s timesteps, which have large initial errors since they are able to model the nominal dynamics but do not reject the perturbations as quickly. Overall, the controllers and timesteps examined were able to successfully stabilize the system about the origin, with the main difference being in the transitory error. Once the systems converged, the error between the different timesteps was due to the integration effects.

For this work, the timesteps used for the integration and control updates range from 0.05 s to 10 s, depending on the specific control approach, with the 10 s timestep only used for control laws developed with the discrete dynamics. Based on the analysis of the uncontrolled dynamics, the maximum integration error is expected to be on the order of 10^{-2} m over an orbit. This is an acceptable level of error for this work and demonstrates that the fixed-step propagation method is able to capture the perturbations in the system. Overall, the integrated solution is shown to be sufficiently accurate and the results from this work are expected to closely mirror the response of an actual spacecraft system.

The controlled results are specific to the continuous and discrete feedback controllers used, but the general trends help understand how the selection of the timestep and dynamic model change the system response. Since the sample-and-hold control implementation necessarily discretizes the control inputs, designing the controller around the discretized solution to the dynamics helps reduce error as the timestep increases. However, not every control law accommodates discretized dynamics, so a continuous control can be developed and approximated by using a sufficiently small timestep for the control update. Differences between the trajectories due to the increase in timestep were seen to be as high as 10^{2} m with the differences decreasing as the timesteps got smaller.

Since this changes the trajectory compared to one generated by a truly continuous control law, it can pose problems in cases where the control law is designed to remain within specified bounds. This impact can be minimized in practice through approaches such as adding a factor of safety to the state bounds or decreasing the timestep. Despite the performance degradation, the control law can still stabilize the system about the origin over a range of timesteps. For the control laws in this work, it is expected that some negative effects will be seen due to the control discretization, but that system convergence can still be achieved by selecting proper tuning parameters and timestep durations.

CHAPTER 4

TUBE-BASED MODEL PREDICTIVE CONTROL

Model predictive control (MPC) is a technique which is based on the optimal control techniques described in Section 2.3.2 but attempts to address some of their shortcomings. In particular, optimal control laws are open-loop where a desired trajectory and control profile are found to satisfy some objective function and set of constraints. However, this ignores system errors, perturbations, and modeling inaccuracies which mean that blindly applying the inputs determined by the optimal control will result in a final system state that does not match the one predicted by the control law. In contrast, MPC repeatedly optimizes over a fixed horizon which results in a closed-loop controller.

In this chapter, a tube-based MPC is developed which guarantees convergence of the system in the presence of a known range of perturbations. This is tested in a simulated orbital environment with J2 and drag perturbations and shown to be successful in maintaining control of a spacecraft.

An overview of both nominal and tube-based MPC is given in Section 4.1 followed by the specific implementation for formation flying in Section 4.2. The simulation framework and results are then presented in Section 4.3.

4.1 Model Predictive Control

MPC works by solving an optimal control problem over some finite-time horizon. The first control of the solution is then applied to the system whereupon the system states are measured and the optimal control problem is solved again from this new state, and the process repeats. Since the time horizon over which the control problem is being solved is continually moving forward, MPC is also known as a Receding Horizon Control (RHC).

4.1.1 Nominal MPC

Consider the discrete, time-invariant, linear system

$$x(k+1) = Ax(k) + Bu(k), \quad k = 0, \ldots, N-1 \tag{4.1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ at each index k. The purpose of MPC is to find a trajectory $x(k)$ and control profile $u(k)$ that satisfies the dynamics, as well as some set of additional constraints, while minimizing an objective function.

The objective function is often comprised of two components, $p : \mathbb{R}^n \to \mathbb{R}$ and $q : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, which can be referred to as the terminal and transient costs, respectively. They are generally used to reduce the error between the $x(k)$ and some desired trajectory and minimize the amount of control used. By adjusting the weights within the terminal and transient cost functions, the behavior of the system can be tuned to better match some desired behavior.

The $\mathcal{X}$ and $\mathcal{U}$ sets define the acceptable bounds of the system to maintain safety and match the available control. Additionally, the terminal set $\mathcal{X}_f$ defines the allowable set of final system states. In the event that $\mathcal{X}_f = 0$, then the terminal cost p becomes redundant. In contrast to standard feedback control, the inclusion of these constraints guarantees that the state and control bounds of any solution are safe and allowable. However, solutions can only be guaranteed for initial conditions which exist within the allowable initial condition set $\mathcal{X}_0 \subset \mathcal{X}$.

Given an initial condition $x(t) \in \mathcal{X}_0$, the MPC formulation can be written in as

$$\begin{aligned} \min_{u} \quad & J(x,u) = p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k \quad k = 0, \ldots, N-1 \\ & x_k \in \mathcal{X},\ k = 0, \ldots, N-1 \\ & u_k \in \mathcal{U},\ k = 0, \ldots, N-1 \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(t) \in \mathcal{X}_0 \end{aligned} \tag{4.2}$$

which finds a solution over the N steps in the planning horizon. The control solution

$$u^*(x(k)) = [u_0^*(x(k)) \ \ldots \ u_{N-1}^*(x(k))] \tag{4.3}$$

can then be applied to the system which results in the trajectory

$$x^*(x(k)) = [x_1^*(x(k)) \ \ldots \ x_N^*(x(k))] \tag{4.4}$$

In practice, the MPC is recalculated repeatedly to generate a closed-loop solution. At each control step, the control solution $u^*(x(k))$ is found and the first control $u_0^*(x(k))$ applied to the system. The state is then measured to obtain a new initial condition $x_0 = x(k+1)$, the MPC is solved again with this new initial condition, and $u_0^*(x(k+1))$ is applied to the system. The closed-loop system can then be described as

$$x(k+1) = Ax(k) + Bu_0^*(x(k)),\ k \geq 0 \tag{4.5}$$

4.1.2 Tube-based MPC

Tube-based MPC [30,55,56] is a modification of the standard MPC that guarantees the system will remain within specified bounds, even when the system is affected by perturbations. In general, there are two ways to approach the problem. One approach [56] generates the region by calculating the reachable set at each point along the time horizon considered by the control law given the current state and possible perturbations. This reachable set is then recalculated each time the control is calculated. A second approach [30], and the one detailed below, generates a single set that contains all possible reachable sets. While less computationally intensive, since it only needs to be precomputed a single time, it does result in a more conservative reachable set.

Tube-based MPC has been used previously in conjunction with spacecraft systems such as in [57] and [58], where it was used in autonomous rendezvous and docking applications. For [57], an adaptive control law is used to generate the reachable sets on-orbit with time-varying dynamics. In contrast, [58] precomputes a time-invariant boundary which is then used at each point along the control law.

Consider a system of the form

$$x(k+1) = Ax(k) + Bu(k) + w(k) \tag{4.6}$$

where $w \in \mathcal{W}$ is the perturbation. The set $\mathcal{W}$ is a compact convex subset of $\mathbb{R}^n$ containing the origin in its interior. Similarly, consider the nominal, unperturbed system

$$\overline{x}(k+1) = A\overline{x}(k) + B\overline{u} \tag{4.7}$$

where $\overline{u}$ is the nominal MPC solution, $u_0(x(k))$, as calculated above.

We now introduce a feedback policy for the perturbed system as

$$u(k) = \bar{u}(k) + K(x(k) - \bar{x}(k)) \tag{4.8}$$

where $K \in \mathbb{R}^{m \times n}$ is a feedback gain.

With this feedback policy, the perturbed system dynamics are

$$x(k+1) = A_D x(k) + B_D \bar{u}(k) + BKe(k) + w(k) \tag{4.9}$$

where $e(k) = x(k) - \bar{x}(k)$ is the error between the perturbed and nominal systems.

Using the error equation, we can write the error dynamics as

$$e(k+1) = A_K e(k) + w(k) \tag{4.10}$$

where $A_K = A_D + B_D K$ and, through the proper selection of K, we can guarantee that A_K is a stable matrix.

With this selection of K, and the resulting A_K, we can now determine how to restrict the boundaries on the nominal MPC so that the original bounds are guaranteed to be maintained. We first define set multiplication as follows [30]: Let $K \in \mathbb{R}^{m \times n}$, then $KA = \{Ka | a \in A\}$. With this, we introduce the uncertainty set $S_K(k)$ as

$$S_K(k) = \sum_{i=0}^{k-1} A_K^k \mathcal{W} \tag{4.11}$$

Similarly, we introduce the set addition function as $A \oplus B = \{a + b | a \in A, b \in B\}$. The titular tube can now be defined as

$$X(i; x(0)) = \{\bar{x}(i)\} \oplus S_K(i), \quad i = 1, \ldots, N \tag{4.12}$$

where $X(i; x(0))$ is the uncertain region at step i, centered on the nominal solution $\bar{x}(i)$. Note that we assume no measurement errors so at the current step $\bar{x}(0)$, the tube is simply the current state $X(0, \bar{x}(0)) = \{\bar{x}(0)\}$.

We now consider how the state and control constraints of the nominal MPC must be constricted to guarantee the perturbed system remains within the allowable bounds. Let $\bar{\mathcal{Z}}$ be the set of state-control constraints for the deterministic system such that $(\bar{x}(k), \bar{u}(k)) \in \bar{\mathcal{Z}}$

for all $k = 0, \ldots, N$ and assume that $\bar{\mathcal{Z}}$ is polytopic. The task is now to find how much the constraints for the uncertain system, $\mathcal{Z}$, must be reduced so that if $(\bar{x}(k), \bar{u}(k)) \in \bar{\mathcal{Z}}$, then $(x(k), u(k)) \in \mathcal{Z}$, for all $k = 0, \ldots, N$.

Let $\mathcal{Z}$ be described as a set of scalar inequalities $cz \leq d$ with $z \in \mathcal{Z}$ or, equivalently, $c_x x + c_u u \leq d$ with $x \in \mathcal{X}$ and $u \in \mathcal{U}$. We introduce an offset θ_i with $i = 0, \ldots, \infty$ defined as

$$\begin{aligned} \theta_i &= \max_e \{c_x e + c_u K e | e \in S_K(i)\} \\ &= \max_w \left\{ \sum_{j=0}^{i-1} c_x A_K^i w | w \in \mathcal{W} \right\} \end{aligned} \tag{4.13}$$

where each element of θ_i is determined individually. Once θ_∞ is found, if $c\bar{z} + \theta_\infty \leq d$ is satisfied for the nominal system, then $cz \leq d$ is satisfied for the perturbed system.

However, since θ_∞ is difficult to compute, the following approximation is used from [30, 59]. Since A_K is stable, for all $\alpha \in (0, 1)$, there exists a finite integer N such that $A_K^N \subset \alpha \mathcal{W}$ and $K A_K^N \mathcal{W} \subset \alpha K \mathcal{W}$. Then,

$$\theta_\infty \leq \theta_N + \alpha \theta_\infty \tag{4.14}$$

or,

$$\theta_\infty \leq (1 - \alpha)^{-1} \theta_N \tag{4.15}$$

The tightened constraints can then be written as

$$c\bar{z} \leq d - (1 - \alpha)^{-1} \theta_N \tag{4.16}$$

Assuming that $\mathcal{W}$ is a box, a suitable N and α can be calculated [30] as

$$\alpha = \max \left(\frac{\max_{w \in W} \|A_K^N w\|_\infty}{\max_{w \in W} \|w\|_\infty}, \frac{\max_{w \in W} \|K A_K^N w\|_\infty}{\max_{Kw \in W} \|w\|_\infty} \right) \tag{4.17}$$

where W is the set of vertices of $\mathcal{W}$. By calculating α in this way for increasing values of N, a suitable $\alpha \in (0, 1)$ can be found.

Once a suitable α and N are found, the upper bound of θ_∞ can be calculated. This, in turn, allows for the calculation of the reduced constraints $\bar{z}$ which can be used in the nominal MPC formulation to calculate the nominal control $\bar{u}^*(x(k))$. Then, at each step, the controller compares the first state of the solution from the previous iteration, $x_1^*(x(k-1))$, to the actual current state $x(k)$ to generate the addition feedback term $K(x(k)-\bar{x}_1^*(x(k-1)))$. The closed-loop, tube-based MPC formulation is then

$$x(k+1) = Ax(k) + B\big[\bar{u}_0^*(x(k)) + K(x(k) - \bar{x}_1^*(x(k-1))\big] \tag{4.18}$$

4.2 Spacecraft Application

One common approach to maintaining a spacecraft's state is to define a volume of space that the spacecraft must remain in and then design control laws that maintain that constraint [60–63]. This application of tube-based MPC is developed from [62] which used an MPC to follow a guidance trajectory. The same guidance and control architecture is used here, but with a guaranteed boundary maintenance using the robust MPC formulation.

When a correction maneuver is required, the guidance law calculates a trajectory that guides the spacecraft from its current state to the nominal, desired orbit. This solution examines a time horizon of around an orbit and a half to allow the guidance law to leverage fuel-efficient maneuver points such as apogee or perigee. This guidance trajectory is then given to the tube-based MPC control which finds a solution that follows the guidance trajectory and rejects perturbations which may occur. The MPC is rerun at each step of the simulation but optimizes over a much shorter time horizon than the guidance trajectory, on the order of tens of minutes.

4.2.1 Guidance Trajectory

We first define the guidance dynamics using the error states. Let the desired trajectory be

$$x_d = [x_d(0), x_d(1), \ldots, x_d(N)] \tag{4.19}$$

such that

$$x_d(k+1) = A_D x_d(k) \tag{4.20}$$

Given the nominal, controlled dynamics as

$$x(k+1) = A_D x(k) + B_D u_g(k) \tag{4.21}$$

the error states $e_g(k) = x(k) - x_d(k)$ can be found to be

$$e_g(k+1) = A_D e_g(k) + B_D u_g(k) \tag{4.22}$$

Note that in this formulation, the bounding constraints on the guidance trajectory can be a time-invariant set which describe the maximum allowable state error relative to the desired trajectory. Since the guidance trajectory is an open-loop control which plans the fuel optimal transfer trajectory, we do not account for perturbations in this portion and these constraints are not adjusted.

The purpose of the guidance law is to compute a fixed time, fuel optimal trajectory that guides the spacecraft from its current state back to the desired orbit. Given the guidance time horizon, N_g, bounds on control inputs, $u_{max}, u_{min} \in \mathbb{R}^n_{>0}$, and bounds on error states $e_{g,max}, e_{g,min} \in \mathbb{R}^m_{>0}$, the guidance error trajectory $e_{g,k} \in \mathbb{R}^n$ and control $u_{g,k} \in \mathbb{R}^m$ can be

calculated as follows:

$$
\begin{aligned}
\min_{x_{g,k},u_{g,k}} \quad & \frac{1}{2}\sum_{k=0}^{N_g-1} \|u_{g,k}\|_1 \\
\text{s.t.} \quad & e_{g,k+1} = A_D e_{g,k} + B_D u_{g,k} \quad k = 0, 1, \ldots, N_g - 1 \\
& e_{g,0} = x(t) - x_d(t) \\
& e_{g,N_g} = 0 \\
& e_{g,k} \leq e_{g,max} \\
& e_{g,k} \geq -e_{g,min} \\
& u_{g,k} \leq u_{g,max} \quad k = 0, 1, 2, \ldots, N_g - 1 \\
& u_{g,k} \geq -u_{g,min} \quad k = 0, 1, 2, \ldots, N_g - 1
\end{aligned}
\tag{4.23}
$$

where $x(t)$ and $x_d(t)$ are the actual and desired states at the current time t. Since the guidance trajectory has an error constraint of $e_{g,N_g} = 0$ for the terminal state and an adjustable time horizon N_g, the guidance law can be considered to be a predefined-time control law. However, in practice, the open-loop guidance law will be unable to reject perturbations which results in the system failing to reach the terminal state.

One consequence of orbital dynamics is that there are points along an orbit where a given maneuver is the most fuel efficient, for example: perigee, apogee, and the intersection of two orbital planes. This has two major implications in the generation of the guidance trajectory. First, a fuel optimal trajectory can experience prolonged periods with significant state errors while it waits for the optimal point to maneuver. This means that if the guidance law were incentivized to minimize the transient state error via a transient error term in the objective function, it would maneuver too early, expending unnecessary control usage. Second, the time horizon over which the guidance law optimizes must be sufficiently long to encompass the fuel optimal points.

The guidance trajectory is only computed once, at the beginning of a course correction maneuver. Once the solution is found, the error trajectory and control profile are saved and used with the tube-based MPC.

4.2.2 MPC Control

The MPC control law is tasked with following the trajectory found by the guidance law despite perturbations. It is built on the optimization formulation shown below in Equation 4.35, which is similar to the guidance law. However, while the guidance law is only run a single time when the spacecraft begins a maneuver, the MPC is re-run each simulation step.

Since an optimal trajectory has already been found by the guidance law, the MPC is able to calculate deviations from the guidance trajectory over a shorter time horizon, N_c. From the stored trajectory and control profile, the MPC selects the period that corresponds to the current interval it is optimizing over. Given the current step of the guidance trajectory that the MPC control law is on, $n_c \in \{1, \ldots, N_g\}$,

$$e_{d,\ell} = e_{g,n} \quad \ell = 0, \ldots, N_c; n = n_c + \ell, \tag{4.24}$$

$$u_{nom,\ell} = u_{g,n} \quad \ell = 0, \ldots, N_c - 1; n = n_c + \ell \tag{4.25}$$

The guidance error trajectory is used as the desired states, $e_{d,\ell}$, while the associated control profile is used as the nominal control inputs, $u_{nom,\ell}$. Note that for any $n > N_g$,

$$e_{d,\ell} = 0_{3\times 1} \tag{4.26}$$

$$u_{nom,\ell} = 0_{3\times 1} \tag{4.27}$$

which represents the spacecraft staying on the drifting, desired orbit after the end of the guidance trajectory.

Looking at the MPC error state dynamics, the total error between the current position and the overall desired trajectory is

$$e(k) = x(k) - x_d(k) \tag{4.28}$$

which is the same definition that is used for the guidance trajectory. Comparing this error to a previously found guidance trajectory error $e_g(i)$ for $i = 1, \ldots, N_g$, gives the MPC error as

$$\begin{aligned} e_m(k) &= e(k) - e_g(k) \\ &= x(k) - x_d(k) - e_g(k) \end{aligned} \tag{4.29}$$

which has dynamics of

$$e_m(k+1) = A_D e_m(k) + B_D(u(k) - u_g(k)) \tag{4.30}$$

where $u(k)$ is the total control applied to the nominal system at step k. For simplicity, we define $u_m(k) = u(k) - u_g(k)$ as the input calculated by the MPC. In other words, the MPC is calculating an adjustment to the guidance control solution to compensate for perturbations.

Similar to the guidance trajectory generation, we can now define a set of time-invariant boundaries for the MPC. While the guidance boundaries are centered around the desired trajectory, the MPC boundaries are centered on the guidance trajectory. This means that when designing the boundary constraints, we have to consider that the guidance error and MPC error can stack such that the total error between the spacecraft's desired state and current state does not exceed the bounds set on the guidance trajectory. This is of note when designing the formation since we have to use the set sum of the two boundary constraints. In the case of "box constraints" where each element is bounded individually, this can trivially be done by element-wise addition of the constraints to find the total keep-in volume. Additionally, the total control constraint is also divided between the guidance and MPC portions so that the sum of the control for the two parts will not be greater than the total available control.

Let us define the total error that the spacecraft must remain within as

$$\begin{aligned} |e| &\leq e_{max} \\ |e| &\geq -e_{min} \end{aligned} \tag{4.31}$$

where $e_{max}, e_{min} \in \mathbb{R}^n_{>0}$ and the absolute value and inequalities are applied elementwise. Then we can set the guidance bounds $e_{g,max}, e_{g,min} \in \mathbb{R}^n_{>0}$ and MPC bounds $e_{m,max}, e_{m,min} \in \mathbb{R}^n_{>0}$ such that

$$\begin{aligned} e_{max} &= e_{g,max} + e_{m,max} \\ e_{min} &= e_{g,min} + e_{m,min} \end{aligned} \tag{4.32}$$

For the tube-based MPC, the $e_{m,max}$ bounds are constricted to generate the bounds $\bar{e}_{m,max}$ which guarantee the spacecraft remains in the $e_{m,max}$ bounds, even in the presence of perturbations.

Similarly, the allowable control bounds are split between the guidance and MPC. Given the total control bounds as

$$\begin{aligned} |u| &\leq u_{max} \\ |u| &\geq -u_{min} \end{aligned} \tag{4.33}$$

with $u_{max}, u_{min} \in \mathbb{R}^m_{>0}$, the guidance and MPC control bounds can be set as

$$\begin{aligned} u_{max} &= u_{g,max} + u_{m,max} \\ u_{min} &= u_{g,min} + u_{m,min} \end{aligned} \tag{4.34}$$

with $u_{g,max}, u_{g,min}, u_{m,max}, u_{m,min} \in \mathbb{R}^m_{>0}$. Again, the $u_{m,max}$ and $u_{m,min}$ constraints for the tube-based MPC are constricted to $\bar{u}_{m,max}$ and $\bar{u}_{m,min}$.

The MPC then optimizes over its time horizon and at each step along the guidance trajectory, a new solution is found as

$$\begin{aligned}
\min_{x,u} \quad & \frac{1}{2} \mid\mid Ru_0 \mid\mid_1 + \frac{1}{2} \sum_{k=1}^{N_c-1} \Big[\mid\mid Qx_{e,k} \mid\mid_1 + \mid\mid Ru_k \mid\mid_1 \Big] \\
\text{s.t.} \quad & e_{k+1} = A_D e_k + B_D u_m \quad k = 0, 1, \ldots, N_c - 1 \\
& e_{m,0} = x(t) - x_d(t) - e_g(t) \\
& e_{m,k} \leq \bar{e}_{m,max} \quad k = 1, 2, \ldots, N_c \\
& e_{m,k} \geq -\bar{e}_{m,min} \quad k = 1, 2, \ldots, N_c \\
& u_{m,k} \leq \bar{u}_{m,max} \quad k = 0, 1, 2, \ldots, N_c - 1 \\
& u_{m,k} \geq -\bar{u}_{m,min} \quad k = 0, 1, 2, \ldots, N_c - 1
\end{aligned} \tag{4.35}$$

Once the control law finds the solution of controls, u_ℓ, that solve the optimization problem, the actual applied control input is determined as

$$u = u_{nom,0} + u_{m,0} + K(e_{m,0} - e_{m,1}(t-1)) \tag{4.36}$$

where K is the tube-based MPC gain, $e_{m,0}$ is the current MPC error, and $e_{m,1}(t-1)$ is the predicted MPC error $e_{m,1}$ from the solution at the previous timestep. This is applied to the system, the new state is then measured, the control law is re-initialized, and the next optimal control is found. The control law runs until the full guidance trajectory has been executed.

The guidance trajectory guarantees that the unperturbed, open-loop system will converge to the desired state in finite-time. Adding the tube-based MPC to reject perturbations reduces the guarantee of the closed-loop system to asymptotic convergence. Since the total allowable error in the system is the combination of the guidance and MPC errors, and since the guidance error is defined to be zero at the conclusion of the guidance trajectory, the final allowable, terminal error for the full system is simply the nominal bounds used to construct the tube-based MPC. Thus, finite-time convergence to a region of the origin is guaranteed with the upper bound of the region determined by the size of the MPC bounds.

4.2.3 Perturbations

In much of the formation flying literature referenced previously [60–62], the existence of perturbing forces is implied but not directly accounted for. Although control laws are developed to reject the perturbing forces, the expected effect of the perturbations on the controllers are only minimally examined.

In [60], the planning error box is introduced as a constriction of the total keep-in boundary to account for the difference in the planning and actual dynamics. To guarantee feasibility for all initial conditions, a scaling variable was introduced into the trajectory optimization problem which allowed the planning error box to grow as needed. However, this negates the purpose of the error boxes since the keep-in boundary is no longer absolute and cannot be used to plan non-overlapping volumes.

In [61, 62], a smaller set is also used to account for the difference in model and actual dynamics. Furthermore, the switching condition to turn on the controller is based on the predicted trajectory of the spacecraft which is designed to initiate correction maneuvers before the spacecraft reaches a state where there is no feasible solution. These methods did not provide guarantees that feasible solutions would always exist, but testing in simulations implied that they were sufficient for the given formation definition.

In general, spacecraft perturbations are well understood but computationally expensive to implement in a spacecraft model. This is particularly relevant when applied to MPC since the optimization process requires rapidly propagating multiple possible spacecraft trajectories. In [46], a list of perturbation sources are given, as well as how the impact of those perturbations varies with a spacecraft's altitude. For Low-Earth Orbits (LEOs), the spacecraft is primarily affected by drag and J2 effects caused by the oblateness of the Earth with other effects due to third-bodies, solar radiation pressure, and other gravitational effects present but at relatively lower levels. By analyzing the anticipated orbit regime, the relevant perturbations can be estimated and used to design the controller to stabilize the system in the presence of those perturbations.

For this work, the only perturbations accounted for are drag and J2 but the approach can be expanded to include any number of desired perturbing forces.

In order to meaningfully calculate the robust boundaries, we first need to determine the set $\mathcal{W}$ of possible perturbations. The perturbations result from the use of the linearized CW dynamics for the guidance and control laws while the actual dynamics for the system include J2 and drag effects. The perturbations are experimentally determined by initializing two spacecraft and propagating them both using the CW dynamics and the full nonlinear dynamics. The the difference in the final states then determines the perturbation. By doing this over multiple initial states which attempt to span the full range of possible orbits, an estimate of the maximum possible perturbations can be determined.

Given an update timestep dt and a corresponding A_D and B_D for the state transition matrix and control input matrix, respectively, we now consider the perturbed MPC dynamics using the error states. Recall that

$$e_m(k) = x(k) - x_d(k) - e_g(k) \tag{4.37}$$

and consider that we use $x(0)$, $x_d(0)$, and $e_g(0)$ to compute $e_m(0)$, all of which are in the LVLH frame. Since the e_g sequence is a reference trajectory and only computed using the CW dynamics, it is unaffected by perturbations. We now find $e_m(1)$ using both the CW dynamics and the perturbed dynamics and compare them to obtain an estimate of the perturbations in the LVLH frame. To do this, we first convert $x(0)$, $x_d(0)$, and $u_m(0)$ to the ECI frame, propagate with the nonlinear dynamics, and then convert back to the LVLH frame, resulting in $x^E(1)$ and $x_d^E(1)$, and then calculate $e_m^E(1)$ as

$$e_m^E(1) = x^E(1) - x_d^E(1) - e_g(1) \tag{4.38}$$

Then, we can propagate e_m using Equation 4.30 as

$$e_m^L(1) = A_D(x(0) - x_d(0)) + B_D u_m(0) - A_D x_g(0) - B_D u_g(0) \tag{4.39}$$

The difference in the two updated states is then the resultant of the perturbation w

$$\begin{aligned} w &= e_m^L(1) - e_m^E(1) \\ &= A_D x(0) + B_D u_m(0) - x^E(1) - A_D x_d(0) + x_d^E(1) \end{aligned} \tag{4.40}$$

This is then calculated over a range of initial conditions expected to span those encountered by the control when in use. The set $\mathcal{W}$ is then chosen as a set that bounds all observed perturbations.

4.2.4 Solution Feasibility

Since the guidance dynamics use the time-invariant error states, feasibility can be proven, given a polytopic initial error set $\mathcal{E}_0^G$ and allowable sets $\mathcal{E}^G$ and $\mathcal{U}^G$ for the allowable errors and controls, respectively.

The guidance trajectory assumes a terminal time t_f which is a function of the discretization timestep and the number of steps N_g in the guidance trajectory. This assures predefined-time convergence of the spacecraft to the desired trajectory for the open-loop, unperturbed system. However, due to the constraints on the guidance trajectory, it must be shown that a feasible solution exists for all $e_0 \in \mathcal{E}_0^G$ given the horizon N_g and allowable sets $\mathcal{E}^G$ and $\mathcal{U}^G$.

Theorem 8. Consider the discrete, time-invariant system

$$x(k+1) = A_D x(k) + B_D u(k) \tag{4.41}$$

with a set of allowable states $\mathcal{X}$, initial conditions $\mathcal{X}_0 \subset \mathcal{X}$, and control constraints $\mathcal{U}$, which are all polytopes that contain 0 in their interior. Now consider the problem of finding a

solution to the problem

$$\begin{aligned} & x_{k+1} = A_D x_k + B_D u_k && k = 0, 1, \ldots, N-1 \\ & x_0 \in \mathcal{X}_0 \\ & x_N = 0 \\ & x_k \in \mathcal{X} && k = 1, \ldots, N \\ & u_k \in \mathcal{U} && k = 0, 1, 2, \ldots, N-1 \end{aligned} \tag{4.42}$$

If a solution exists for each vertex of $\mathcal{X}_0$, then the problem is feasible for all $x_0 \in \mathcal{X}_0$.

Proof. First note that the equality constraints can be stacked to obtain a single equality constraint

$$\begin{bmatrix} -I & 0 & \ldots & 0 & 0 & B & 0 & \ldots & 0 \\ A & -I & \ldots & 0 & 0 & 0 & B & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & A & -I & 0 & 0 & \ldots & B \\ 0 & 0 & \ldots & 0 & I & 0 & 0 & \ldots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \\ u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} -A_D x_0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{4.43}$$

and since it is a linear system of equations, any convex combination of solutions is also a solution. Since $\mathcal{X}_0$ is a polytope, any point in the set can be expressed as a convex combination of the vertices where the coefficients are non-negative and sum to 1. Furthermore, since the state constraints $\mathcal{A}$ and control constraints $\mathcal{U}$ are convex sets, convex combinations of each x_k and u_{k-1}, $k = 1, \ldots, N$ by those same coefficients will remain in their respective sets. Thus, if a feasible solution exists for each vertex of $\mathcal{X}_0$, a feasible solution exists for all $x_0 \in \mathcal{X}_0$. $\square$

MPC generally solves the optimization problem at regular intervals with updated initial conditions at each step along the simulation. With standard MPC, this is necessary so that the MPC can measure the errors cause by perturbations and recalculate the control to close the loop. In that case, feasibility must be assured for each subsequent step along the system's closed-loop trajectory.

However, with tube-based MPC, the feedback gain fulfills that role of rejecting perturbations. Thus, tube-based MPC can solve the optimization problem for the initial conditions and find an open-loop solution. Recall that the selection of the feedback gain and the constriction on the bounds for the tube-based MPC problem are selected based on the expected perturbations such that the system is guaranteed to not exceed the overall allowable bounds. The system can then follow the solution and use the feedback gain to reject perturbations based on the difference between the solution and the actual trajectory without needing to repeatedly solve the optimization problem. While the system may exceed the constricted tube-based MPC bounds due to perturbations, the feedback gain keeps the system within the overall desired bounds. Thus, feasibility only needs to be proven for the initial condition.

In practice it can still be beneficial for the tube-based MPC to recalculate solutions periodically based on the updated system states. This allows the MPC to design fuel-efficient trajectories that account for the experienced perturbations instead of solely relying on the feedback gain. However, as a result of the perturbations, subsequent steps may find the system to be outside the allowable, constricted bounds used by the tube-based MPC and result in an infeasible problem. In these situations, an updated solution cannot be generated, but the previous solution the system has been following until this point is still valid. The system can continue to use this previous solution until the boundary conditions are satisfied to generate a new, updated solution. Thus, feasibility only needs to be guaranteed for the initial tube-based MPC formulation and for subsequent states a new solution can be generated, if possible, or the previous solution can continued to be followed.

Applying the theorem shows that feasible solutions can be guaranteed for both the guidance and control portions of the formation flying architecture by checking the vertices

of the initial conditions. This approach does have some limitations, however. First, the boundary constraints must be polytopic sets with a finite number of vertices which eliminates bounding sets defined by spheres or ellipses. Second, the computation time needed increases proportionally to the number of vertices used in the bounding set. However, both these limitations can be mitigated by using a simpler set, of which the desired constraints are a subset, to check feasibility. Then, if the problem is feasible for this set, the problem is guaranteed to be feasible for the actual bounds.

Assuming that there is a feasible trajectory, we now show that an optimal trajectory exists. This is done by applying the Weierstrass Theorem which places conditions on the objective function f and constraint set $\mathcal{X}$.

Theorem 9. [Weierstrass Theorem [32, Theorem 3.3]] If f is continuous and $\mathcal{X}$ is closed and bounded, then f attains a minimum and maximum on $\mathcal{X}$.

Applying this to the guidance trajectory problem, the objective function is

$$J_G = \frac{1}{2} \sum_{k=0}^{N_g-1} \|u_{g,k}\|_1 \tag{4.44}$$

which is continuous and the constraints are closed and bounded. Thus, if a feasible guidance solution exists, there is an optimal solution. Similarly, the MPC objective function is

$$J_{MPC} = \frac{1}{2} \sum_{k=1}^{N_g} \|e_{m,k}\|_1 + \frac{1}{2} \sum_{k=0}^{N_g-1} \|u_{m,k}\|_1 \tag{4.45}$$

which is also continuous and satisfies the Weierstrass Theorem. Additionally, the constraint sets for the state errors and control usage are closed and bounded for both the guidance and MPC problems. Thus, if the guidance and MPC problems are feasible, then there is an optimal solution to each.

4.3 Results

This section details the application of tube-based MPC to the formation flying architecture used in [62]. Since the architecture itself was already demonstrated, this portion only demonstrates that the tube-based MPC approach is sufficient to keep a spacecraft within the allowable bounds. When applied to a full spacecraft formation, the same approach can be applied to each spacecraft in the formation.

We first estimate the bounds of the possible perturbations based on the anticipated errors. These perturbations are then used to determine the constricted bounds for MPC which will guarantee that the spacecraft will remain in the allowable bounds despite perturbations. The controller is then tested using the CW dynamics with perturbations added, as well as using the full nonlinear dynamics with J2 and drag perturbations.

4.3.1 Perturbation Calculation

The perturbations were estimated using the method in 4.2.3. For each perturbation sample, a reference orbit was selected as

$$x_{ref}^{EL} = \begin{bmatrix} a \in U(6\,728\,000\,\text{m}, 7\,378\,000\,\text{m}) \\ e \in U(0, 0.01) \\ i \in U(0°, 90°) \\ \Omega \in U(0°, 360°) \\ \omega \in U(0°, 360°) \\ \nu \in U(0°, 360°) \end{bmatrix} \tag{4.46}$$

where x_{ref}^{EL} is the initial reference state in Keplerian coordinates, semi-major axis a, eccentricity e, inclination i, right ascension of the ascending node Ω, argument of perigee ω, and true anomaly ν. This state is then converted to ECI coordinates.

The sample state used for x_d is then chosen in the LVLH frame relative to the reference as

$$x_d = \begin{bmatrix} U(-1000\,\mathrm{m}, 1000\,\mathrm{m}) \\ U(-1000\,\mathrm{m}, 1000\,\mathrm{m}) \\ U(-1000\,\mathrm{m}, 1000\,\mathrm{m}) \\ U(-50\,\mathrm{m/s}, 50\,\mathrm{m/s}) \\ U(-50\,\mathrm{m/s}, 50\,\mathrm{m/s}) \\ U(-50\,\mathrm{m/s}, 50\,\mathrm{m/s}) \end{bmatrix} \tag{4.47}$$

The sample state used for x is chosen as

$$x = x_d + \begin{bmatrix} U(-1000\,\mathrm{m}, 1000\,\mathrm{m}) \\ U(-1000\,\mathrm{m}, 1000\,\mathrm{m}) \\ U(-1000\,\mathrm{m}, 1000\,\mathrm{m}) \\ U(-10\,\mathrm{m/s}, 10\,\mathrm{m/s}) \\ U(-10\,\mathrm{m/s}, 10\,\mathrm{m/s}) \\ U(-10\,\mathrm{m/s}, 10\,\mathrm{m/s}) \end{bmatrix} \tag{4.48}$$

with the control input chosen as

$$u = \begin{bmatrix} U(-0.25\,\mathrm{m/s^2}, 0.25\,\mathrm{m/s^2}) \\ U(-0.25\,\mathrm{m/s^2}, 0.25\,\mathrm{m/s^2}) \\ U(-0.25\,\mathrm{m/s^2}, 0.25\,\mathrm{m/s^2}) \end{bmatrix} \tag{4.49}$$

The possible x_d states cover the expected range of desired relative orbits while the x state covers the expected range of allowable error. The possible u values represent the maximum allowable controls that can be added to the system.

After sampling 50 000 points, the maximum magnitude for the perturbations along each axis were found to be

$$w_{max} = \begin{bmatrix} 0.071\,13\,\text{m} \\ 0.072\,76\,\text{m} \\ 0.017\,42\,\text{m} \\ 0.019\,09\,\text{m/s} \\ 0.017\,63\,\text{m/s} \\ 0.000\,51\,\text{m/s} \end{bmatrix} \tag{4.50}$$

For simplicity, and to account for the fact that this is only a sample-based estimate, the perturbation set used to design the Tube-based MPC is expanded slightly to

$$\mathcal{W} = \left\{ w \mid |w| \leq \begin{bmatrix} 0.1\,\text{m} & 0.1\,\text{m} & 0.1\,\text{m} & 0.02\,\text{m/s} & 0.02\,\text{m/s} & 0.02\,\text{m/s} \end{bmatrix}^T \right\} \tag{4.51}$$

where the absolute value and inequality are applied element-wise.

4.3.2 Allowable Bounds

The bounds for each portion of the control architecture are as presented in Table 4.1. The overall design constraint for this work was chosen to keep the spacecraft within 500 m of the desired trajectory along each axis. A velocity constraint of 10 m/s difference along each axis was also selected. This overarching constraint was then divided between the guidance and MPC portions with the guidance constraints set at 425 m and MPC set at 75 m with the velocity error divided between them at 5 m/s each.

With the estimation of the perturbations and the nominal bounds for the MPC decided, the robust MPC bounds can be calculated. The K feedback was selected so that the closed-loop, discrete system would have the poles indicated in Table 4.2. Then, applying Equations 4.13, 4.16, and 4.17 with N selected to be 250, resulted in $\alpha = 1.160\,16 \times 10^{-4}$

Table 4.1: Bounds for each portion of the control architecture. Norms and inequalities are all elementwise.

Total	$\lvert e\rvert \leq \begin{bmatrix} 500\,\text{m} \\ 500\,\text{m} \\ 500\,\text{m} \\ 10\,\text{m/s} \\ 10\,\text{m/s} \\ 10\,\text{m/s} \end{bmatrix}$	Guidance	$\lvert e_0\rvert \leq \begin{bmatrix} 350\,\text{m} \\ 350\,\text{m} \\ 350\,\text{m} \\ 1\,\text{m/s} \\ 1\,\text{m/s} \\ 1\,\text{m/s} \end{bmatrix}$	MPC	$\lvert e_0\rvert \leq \begin{bmatrix} 20\,\text{m} \\ 20\,\text{m} \\ 20\,\text{m} \\ 0.25\,\text{m/s} \\ 0.25\,\text{m/s} \\ 0.25\,\text{m/s} \end{bmatrix}$
	$\lvert e\rvert \leq \begin{bmatrix} 500\,\text{m} \\ 500\,\text{m} \\ 500\,\text{m} \\ 10\,\text{m/s} \\ 10\,\text{m/s} \\ 10\,\text{m/s} \end{bmatrix}$		$\lvert e\rvert \leq \begin{bmatrix} 425\,\text{m} \\ 425\,\text{m} \\ 425\,\text{m} \\ 5\,\text{m/s} \\ 5\,\text{m/s} \\ 5\,\text{m/s} \end{bmatrix}$		$\lvert e\rvert \leq \begin{bmatrix} 60\,\text{m} \\ 30\,\text{m} \\ 40\,\text{m} \\ 4\,\text{m/s} \\ 4\,\text{m/s} \\ 4\,\text{m/s} \end{bmatrix}$
	$\lvert u\rvert \leq \begin{bmatrix} 0.02\,\text{m/s}^2 \\ 0.02\,\text{m/s}^2 \\ 0.02\,\text{m/s}^2 \end{bmatrix}$		$\lvert u\rvert \leq \begin{bmatrix} 0.01\,\text{m/s}^2 \\ 0.01\,\text{m/s}^2 \\ 0.01\,\text{m/s}^2 \end{bmatrix}$		$\lvert u\rvert \leq \begin{bmatrix} 0.006\,\text{m/s}^2 \\ 0.006\,\text{m/s}^2 \\ 0.006\,\text{m/s}^2 \end{bmatrix}$

and

$$\bar{e}_{m,max} = \bar{e}_{m,min} = \begin{bmatrix} 61.278277084557388 \\ 32.244247139161274 \\ 43.980983091803651 \\ 4.866698144017604 \\ 4.774377684097654 \\ 4.815934382031404 \end{bmatrix} \tag{4.52}$$

$$\bar{u}_{m,max} = \bar{u}_{m,min} = \begin{bmatrix} 0.006895504931198 \\ 0.007028254091638 \\ 0.007329580334443 \end{bmatrix}$$

For simplicity, these bounds were further restricted to convenient values, as shown in Table 4.1.

Finally, the bounds for the initial conditions e_0 were selected and verified by testing that for each vertex of the initial conditions, there is a feasible solution that exists within the allowable bounds and which terminates at the origin. The initial bounds are separated for the

guidance and MPC portions, with the guidance initial conditions representing the maximum error between the guidance initial condition and the desired trajectory. Similarly, the MPC initial error is the largest initial error between the MPC initial condition and the initial state of the guidance trajectory. Due to the use of the error states in the problem formulation, the guidance and MPC feasibility were able to be verified independent of each other. While the initial condition bounds shown in Table 4.1 satisfy the feasibility constraints, there is no guarantee that these are the largest possible sets that guarantee feasibility.

4.3.3 Perturbed CW Dynamics

We first test the robust control formulation using the CW dynamics and manually adding random perturbations that exist within $\mathcal{W}$. The parameters used in this simulation are found in Table 4.2.

Table 4.2: Parameters used in the simulation

dt	10 s
N_g	800 steps
N_m	30 steps
n	0.0011 1/s
Q_{mpc}	0.001 I
R_{mpc}	I
Poles	$\begin{bmatrix}0.85 & 0.85 & 0.95 & 0.95 & 0.9 & 0.9\end{bmatrix}$
$\mathcal{W}$	$\lvert w\rvert \leq \begin{bmatrix}0.1\,\text{m} & 0.1\,\text{m} & 0.1\,\text{m} & 0.02\,\text{m/s} & 0.02\,\text{m/s} & 0.02\,\text{m/s}\end{bmatrix}^T$

For this, the desired trajectory x_d is set to be the origin over the full duration of the guidance law. The initial state for the guidance trajectory is randomly selected from one of the vertices that bounds the possible initial guidance states (Table 4.1). Once the guidance trajectory is planned, the initial state is then perturbed such that the initial MPC error $e_m(0)$ lies on a random vertex of the allowable MPC initial bounds. At each step, the spacecraft state is updated as

$$x(i+1) = A_D x(i) + B_D u(i) + w(i) \tag{4.53}$$

where $w(i)$ is randomly chosen such $w(i) \in U(-1,1)*\mathcal{W}$ where U is the uniform distribution. For discretizing the system, a timestep of 10 s and a mean motion of 0.0011 1/s were used. The control gain K in the Tube-MPC feedback gain was selected to give the discrete system the poles are indicated in Table 4.2.

A total of 50 simulations are run with the initial conditions randomly selected from the bounding vertices. In Figure 4.1, the guidance trajectories for all the simulations are shown and in Figure 4.2, the control profiles are shown. Note that in all cases, the states and controls remain within the allowable bounds. The final error for all the states averages zero across all the simulations but, due to the perturbations, each individual simulation has some non-zero error. Looking at the control profiles, the large spikes result from the guidance control as it tends towards using impulsive-like controls in the fuel-optimal locations. Additionally, each axis also has a pattern of smaller controls which come from the tube-based MPC attempting to reject the perturbations.

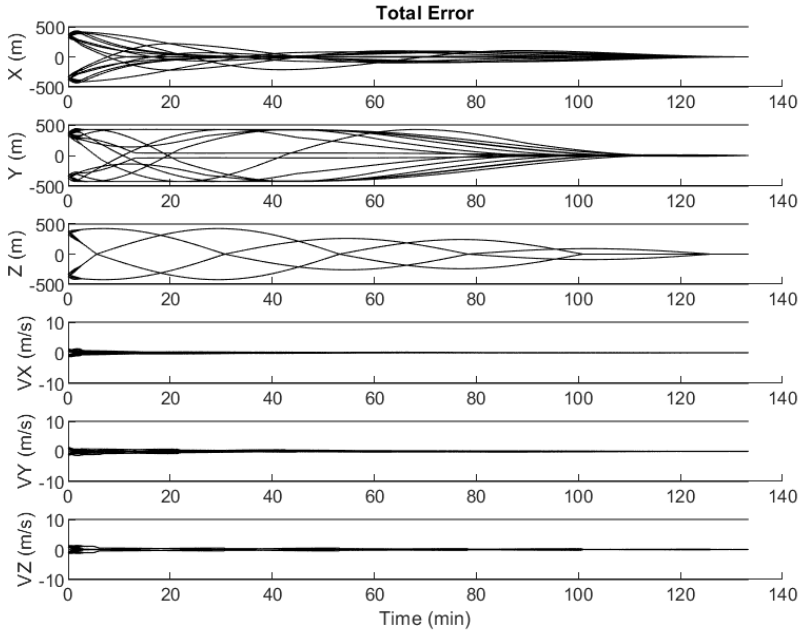

Fig. 4.1: Error trajectory calculated for guidance and MPC law over 50 randomized simulations.

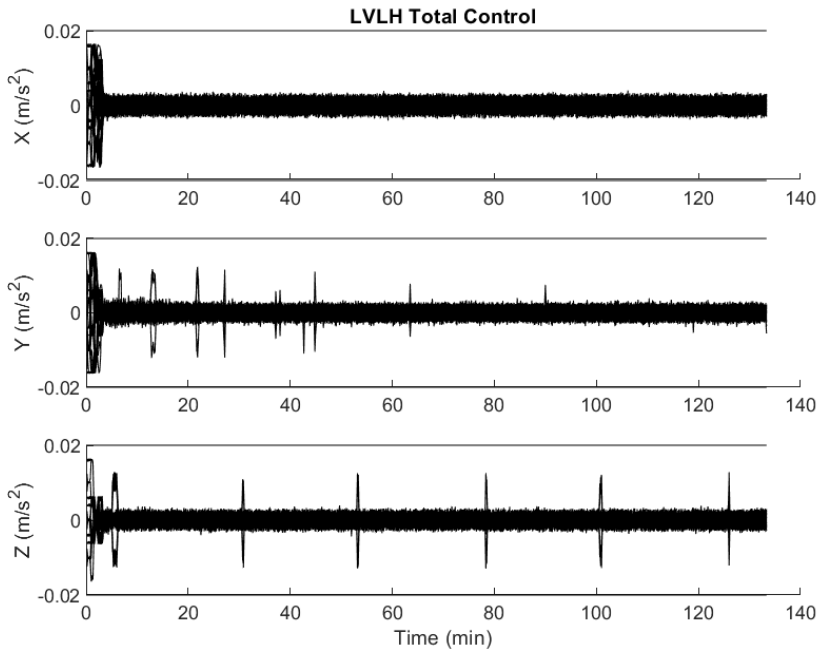

Fig. 4.2: Total control acceleration, consisting of guidance, MPC, and feedback, over 50 simulations.

In order to better examine the terminal behavior and overall convergence, the position errors and control magnitudes are presented on a log scale. In Figures 4.3 and 4.4, the position errors for the in-plane subsystem and cross-track subsystem are shown, respectively. For the in-plane errors, they show a gradual decrease until they converge to approximately 10^{-1} m and then exhibit chattering behavior. The convergence times have some variation but overall, each converges within the convergence time bound, and the bound does not appear to be overly conservative.

While the cross-track subsystem (Figure 4.4) also is able to converge within the allowed bounds, the error trajectories are quite different. In general, only two main trajectories are visible, although some minor deviations exist within those, as opposed to the wider range of trajectories seen with the in-plane dynamics. Additionally, the system shows very distinct oscillations before converging on the origin where chattering occurs.

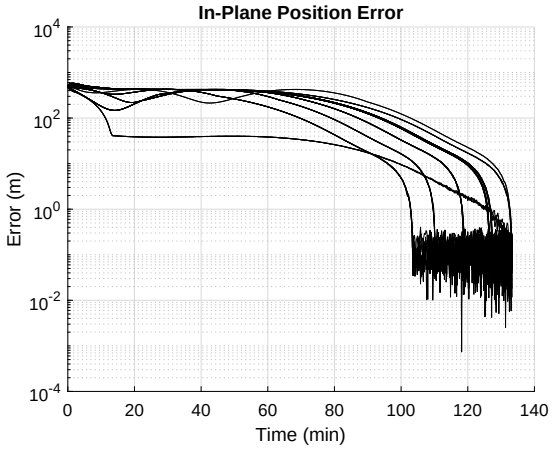

Fig. 4.3: In-plane position errors for the CW dynamics using the guidance and tube-based MPC.

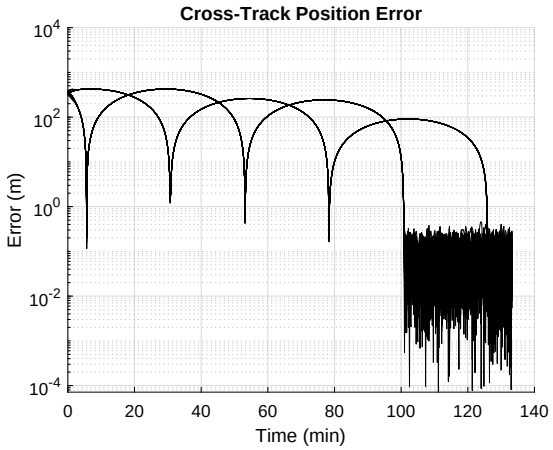

Fig. 4.4: Cross-track position errors for the CW dynamics using the guidance and tube-based MPC.

In Figures 4.5 and 4.6, the control magnitudes for the two subsystems are shown. Some larger peaks in the control trajectories can be seen, but overall, the controls for both subsystems exhibit large amounts of chatter and a very non-smooth trajectory.

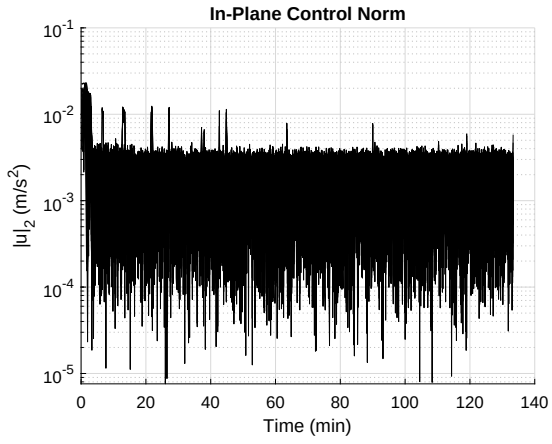

Fig. 4.5: In-plane control for the CW dynamics using the guidance and tube-based MPC.

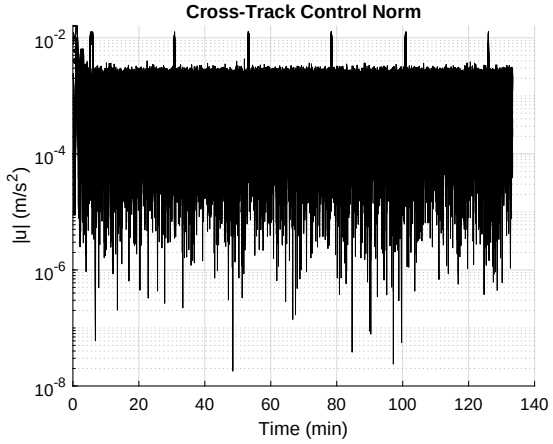

Fig. 4.6: Cross-track control for the CW dynamics using the guidance and tube-based MPC.

This demonstrates that the guidance and control algorithm is able to successfully control the spacecraft when the simulation dynamics match the model dynamics and the pertur-

bations are within the bounds used for designing the controller. One item of note is that many of the constraints appear to be poorly fit for the system. For example, examining the allowable bounds for the velocities show that the actual velocities never approach the bounds. While this is not necessarily detrimental to the system, it does suggest that the bounds can be adjusted to boundaries more relevant to a desired application. Additionally, the total control never reaches the control boundary while the guidance control does occasionally hit its maximum control boundary. This suggests that there may be benefit in reducing the amount of control available to the MPC and increase that available to the guidance trajectory.

4.3.4 Two-Body Dynamics

Once the control was shown to work with the perturbed CW dynamics, it was tested using the two-body, nonlinear dynamics. The General Mission Analysis Tool (GMAT) [64] was used to simulate the dynamics via the MATLAB API with J2 and drag perturbations. A single reference orbit and desired relative orbit (Table 4.3) was used for all simulations. For each simulation, a guidance trajectory was found that corresponded to a random vertex of the allowable boundary volume. Once a guidance trajectory was found, the state was perturbed so that the initial condition corresponded to a random vertex of the allowable MPC error. The simulation then ran for the duration of the guidance trajectory with the control at each step being calculated from the guidance control, MPC control, and feedback control and then propagated to the next step using the nonlinear dynamics. The parameters for the controls are identical to those used for the perturbed linear dynamics (Table 4.2) while the physical parameters used for the spacecraft are given in Table 4.4.

In Figures 4.7 and 4.8, the total errors for in-plane and cross-track subsystems, respectively, are shown for 50 randomly initialized simulations. Overall, the plots look similar to the CW results, with the in-plane error smoothly converging to the origin and the cross-track showing oscillations. However, at the end of the simulations there is significantly less chattering and the average errors are lower. This indicates that the perturbation estimate used with the CW dynamics was overly conservative.

Table 4.3: Reference orbit in Keplerian coordinates and desired orbit in ROEs.

Reference Orbit	$a = 6\,878\,000\,\text{m}$ $e = 1 \times 10^{-4}$ $i = 170°$ $\Omega = 126°$ $\omega = 0°$ $\nu = 0°$
Relative Orbit	$\delta a = 0$ $\delta\lambda = 0.1$ $\delta e_x = 0.000\,145\,4$ $\delta e_y = 0$ $\delta i_x = 0.000\,145\,4$ $\delta i_y = 0$

Table 4.4: Spacecraft parameters used for the nonlinear spacecraft propagation.

Parameter	Value
Initial Wet Mass	$24\,\text{kg}$
I_{SP}	$160\,\text{s}$
Projected Surface Area	$0.048\,\text{m}^2$
C_D	2.2

The tube-based MPC only guarantees that the position error at the end of the guidance trajectory will be within the nominal bound used with the MPC. This sets an upper bound on the terminal position error. However, for both the in-plane and cross-track results, the errors are much less than this upper bound showing that in practice, closer convergence to the origin can be achieved.

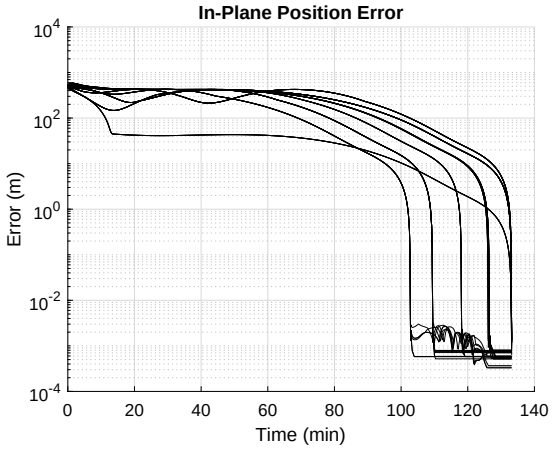

Fig. 4.7: In-plane position errors for the two-body dynamics using the guidance and tube-based MPC.

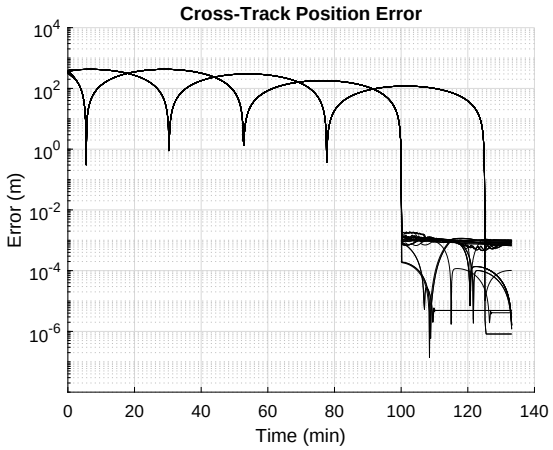

Fig. 4.8: Cross-track position errors for the two-body dynamics using the guidance and tube-based MPC.

In Figures 4.9 and 4.10, the corresponding plots are shown for the in-plane and cross-track controls, respectively, with the two-body dynamics. In contrast to the perturbed CW dynamics, the in-plane controls now have a visible overall trajectory. There are large peaks in the control from the guidance solution, but otherwise the control profile is relatively smooth as it rejects perturbations. However, there are still some chattering effects, especially at the end of the simulations. The cross-track control is smoother than its counterpart that was applied to the perturbed CW model (Figure 4.4) but less smooth than the in-plane control. Some of this may result from the fact that the average control magnitude is roughly an order of magnitude smaller than the in-plane control, so small chattering effects are magnified. However, the control still follows a similar pattern, where the guidance solution results in spikes in control usage to mimic impulsive maneuvers, and the MPC applies smaller amounts of control to reject perturbations.

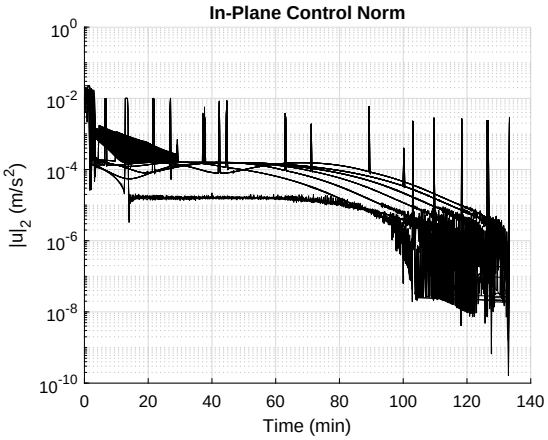

Fig. 4.9: In-plane control for the two-body dynamics using the guidance and tube-based MPC.

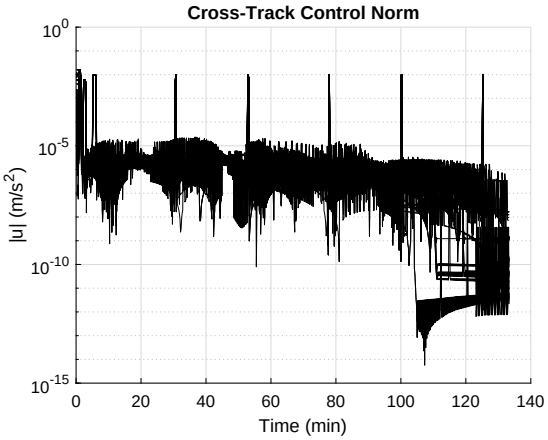

Fig. 4.10: Cross-track control for the two-body dynamics using the guidance and tube-based MPC.

Overall, this control usage is higher than what was seen in [62], despite the keep-in volumes being roughly similar sizes. This appears to result from the switching condition used in [62] which initiated a correction maneuver based on estimating the future trajectory of the spacecraft. This means that the actual initial conditions for the guidance and MPC laws for the nominal approach are actually smaller than the designated boundaries. In contrast, the simulations in this analysis use all initial conditions which lie on the boundary of the allowable space. Additionally, the tube-based MPC velocity constraints are much larger than what were seen in [62]. While this allows for greater flexibility in finding guidance and MPC solutions, it also increases the control needed to return to the desired trajectory.

4.4 Conclusion

This chapter details the use of tube-based MPC to develop a robust controller that is proven to stabilize the spacecraft system despite the presence of nonlinear dynamics and perturbing forces. A guidance law was used to find a fuel optimal trajectory that returns the spacecraft back to the desired state within a fixed time horizon. An MPC is then used to follow that trajectory and reject perturbations. The maximum possible perturbations were estimated by comparing the linear CW dynamics to the nonlinear dynamics and then

used to constrict the boundaries on the MPC. When used in conjunction with a feedback control, the restricted bounds guarantee that the perturbed system will remain within the nominal desired boundary while not exceeding the allowable control. Thus, while finite-time convergence is not guaranteed in the presence of perturbations, the closed-loop control does guarantee finite-time convergence to a known bound.

The guidance and control laws were tested using both perturbed linear dynamics and nonlinear dynamics. In both cases, the system was able to converge on the desired state and reject perturbations. Control usage was higher, however, for the perturbed linear dynamics, indicating that the estimated perturbing forces were a conservative estimate. The guidance and control laws can easily be fit into the formation flying architecture detailed in [62], allowing for robust stability of the spacecraft formation.

CHAPTER 5

FINITE-TIME CONTROL VIA COMMON LYAPUNOV FUNCTIONS

Control using common Lyapunov functions has been used to guarantee stability of switched and hybrid systems [65–68]. The use of common Lyapunov functions was extended in [31] to define a controller that ensures finite-time stability for single-input LTI systems in Brunovsky form. The design and tuning of the controller, as applied to the relative orbit maintenance problem, are explored in this chapter.

5.1 Common Lyapunov Theory

Consider a system of the form

$$\dot{x} = Ax + Bu \tag{5.1}$$

where $A \in \mathbb{R}^{n\times n}$ and $B^{n\times 1}$ are in the Brunovsky form

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{5.2}$$

Let $K \in \mathbb{R}^{1\times n}$ be a feedback gain such that the closed-loop matrix $A_s = A + BK$ is Hurwitz.

We also introduce a function $\delta : \mathbb{R}_{>0} \to \mathbb{R}^{n\times n}$ as

$$\delta(T) = \begin{bmatrix} T^{-n} & 0 & \dots & 0 \\ 0 & T^{-n+1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & T^{-1} \end{bmatrix} \tag{5.3}$$

Note that with δ, we can make the following connections:

$$\begin{aligned} \delta(T)A\delta(T)^{-1} &= T^{-1}A \\ \delta(T)B &= T^{-1}B \\ \frac{d}{dT}\delta(T) &= T^{-1}M\delta(T) \end{aligned} \tag{5.4}$$

where M is defined to be

$$M = \begin{bmatrix} -n & 0 & \dots & 0 \\ 0 & 1-n & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -1 \end{bmatrix} \tag{5.5}$$

and is noted to be a stable matrix. The connection between the (A, B) system and the M system is shown by finding the dynamics of $y = \delta(T)x$ as

$$\dot{y} = T^{-1}(Ay + Bu + \dot{T}My) \tag{5.6}$$

We can now introduce the main results presented in [31] as follows.

Conjecture 1 ([31, Theorem 1])**.** There exists a gain K such that $A_s = A + BK$ is stable and A_s and M posses a common quadratic Lyapunov function. This is of the form

$$\begin{aligned} QA_s + A_s^T Q &= -P \\ QM + MQ &= -R \end{aligned} \tag{5.7}$$

where Q, P, and R are symmetric positive definite matrices.

Theorem 10 ([31, Theorem 2])**.** Suppose that $V = x^T Qx$ is a common quadratic Lyapunov function for two stable matrices A_s and M. Define a function $T : \mathbb{R}^n \to \mathbb{R}$ such that $T(0) = 0$ and

$$x^T \delta(T) Q \delta(T) x = 1 \tag{5.8}$$

for $x \neq 0$. Then, $u = K\delta(T(x))x$ brings any state for the system (A, B) to the origin in finite time.

5.1.1 Control Analysis

From the control law, we can now determine relationships between system performance and the Q, P, and R matrices.

The rate of change of T is found to be [31]

$$\dot{T} = -\frac{x^T\delta(T)P\delta(T)x}{x^T\delta(T)R\delta(T)x} \tag{5.9}$$

Since P and R are, by definition, positive-definite matrices, the T value will always be decreasing. For convenience, let $y = \delta(T)x$. Then, finding the magnitude of $\dot{T}$ gives

$$\|\dot{T}\| = \frac{\|y^TPy\|}{\|y^TRy\|} \tag{5.10}$$

Since P and R are symmetric, positive-definite matrices,

$$\begin{aligned} \lambda_{min}(P)\|y\|^2 \leq \|y^TPy\| \leq \lambda_{max}(P)\|y\|^2 \\ \lambda_{min}(R)\|y\|^2 \leq \|y^TRy\| \leq \lambda_{max}(R)\|y\|^2 \end{aligned} \tag{5.11}$$

where $\lambda_{min}(\cdot)$ and $\lambda_{max}(\cdot)$ represent the minimum and maximum eigenvalues of the indicated matrix, respectively. With this, the bounds on $\dot{T}$ are independent of y and can be found as

$$\frac{\lambda_{min}(P)}{\lambda_{max}(R)} \leq \|\dot{T}\| \leq \frac{\lambda_{max}(P)}{\lambda_{min}(R)} \tag{5.12}$$

Given an initial T_0 at time $t_0 = 0$, the time t_f that the system will reach $T = 0$ is bounded as

$$T_0\frac{\lambda_{min}(R)}{\lambda_{max}(P)} \leq t_f \leq T_0\frac{\lambda_{max}(R)}{\lambda_{min}(P)} \tag{5.13}$$

Similarly, the bound on the control can be examined as

$$\begin{aligned} \|u\| &= \|Ky\| \\ &\leq \|K\|\|y\| \end{aligned} \tag{5.14}$$

With y constrained such that $y^TQy = 1$, the bounds on y can be found as

$$\begin{aligned} \|y^TQy\| &= 1 \\ \lambda_{min}(Q)\|y\|^2 \leq \|y^TQy\| &\leq \lambda_{max}(Q)\|y\|^2 \\ \lambda_{min}(Q) \leq \frac{1}{\|y\|^2} &\leq \lambda_{max}(Q) \\ \frac{1}{\sqrt{\lambda_{min}(Q)}} \geq \|y\| &\geq \frac{1}{\sqrt{\lambda_{max}(Q)}} \end{aligned} \tag{5.15}$$

The upper bound on u can then be written as

$$\|u\| \leq \frac{\|K\|}{\sqrt{\lambda_{min}(Q)}} \tag{5.16}$$

Note that this is a global bound, independent of the current state x, due to the $y^TQy = 1$ constraint.

This analysis gives some insight into how Q, P, and R can be tuned in order to influence the system behavior. By selecting matrices with appropriate eigenvalues, the bounds on the convergence time and maximum control can be shifted. However, the three matrices cannot be selected independently due their relationships via the common Lyapunov constraint.

5.1.2 Parameter Tuning

From the analysis on the bounds for the convergence time and maximum control, the key parameters are identified to be $\lambda_{min}(Q)$, $\lambda_{max}(R)$, and $\lambda_{min}(P)$. Specifically, by increasing $\lambda_{min}(Q)$, the maximum control bound is decreased, and by decreasing the ratio of the $\lambda_{max}(R)$ to $\lambda_{min}(P)$, the final time is minimized. In order to adjust these variables, a semidefinite program [69] is used to maintain the common Lyapunov constraint as a linear matrix inequality (LMI) while pushing the eigenvalues of Q, P, and R in the desired

directions. The problem is formulated as

$$
\begin{aligned}
\min_{q_{min}, r_{max}, p_{min}} \quad & -q_w q_{min} + r_w r_{max} - p_w p_{min} \\
\text{s.t.} \quad & QA_s + A_s^T Q = -P \\
& QM + MQ = -R \\
& q_{min} I \leq Q \leq q_{ub} I \\
& p_{min} I \leq P \leq p_{ub} I \\
& r_{lb} I \leq R \leq r_{max} I \\
& q_{min} \geq q_{lb} \\
& p_{min} \geq p_{lb} \\
& r_{max} \leq r_{ub}
\end{aligned}
\tag{5.17}
$$

where $q_{ub}, p_{ub}, r_{ub} \in \mathbb{R}_{>0}$ are the upper allowable bounds, $q_{lb}, p_{lb}, r_{lb} \in \mathbb{R}_{>0}$ are the lower allowable bounds, and $q_w, p_w, r_w \in \mathbb{R}_{\geq 0}$ are weights. The lower bounds on Q, P, and R drive the smallest eigenvalue of the respective matrix while the upper bounds constrain the maximum eigenvalue of each. The objective function is minimized by increasing the minimum eigenvalues of Q and P and decreasing the maximum eigenvalue of R, which drives the bounds on the maximum control and convergence time towards the desired behavior.

5.2 Results

The common Lyapunov control parameters are explored using the CW dynamics with single-input in-plane controls to determine how effective the control and convergence time bounds are. Multiple controllers are developed using the parameter tuning method with varying acceptable ranges for the common Lyapunov matrices, which are then applied to the CW dynamics. From the results, the correlation between the theoretical bounds and the actual performance is compared. A single controller with suitable behavior is then selected and applied to the perturbed two-body dynamics.

Table 5.1: Parameters used in Equation 5.17 for the optimization of the control law.

Parameter	Value
q_{ub}	$\begin{bmatrix}10^{-1} & \dots & 10^{2}\end{bmatrix}$
p_{ub}	1×10^{5}
r_{ub}	1×10^{5}
q_{lb}	$\begin{bmatrix}10^{-3} & \dots & 10^{1}\end{bmatrix}$
p_{lb}	$\begin{bmatrix}10^{-3} & \dots & 10^{1}\end{bmatrix}$
r_{lb}	$\begin{bmatrix}10^{-3} & \dots & 10^{1}\end{bmatrix}$
α	$\begin{bmatrix}0 & \dots & 1\end{bmatrix}$
q_w	1
p_w	α
r_w	$1 - \alpha$

5.2.1 Parameter Selection

The CW dynamics were divided into the single-input in-plane and cross-track subsystems and transformed into the Brunovsky form. For each subsystem, a controller was designed using Equation 5.17 and the system was simulated and analyzed to determine the practical impact of the Q, P, and R eigenvalues.

For each simulation, the stabilizing feedback gain K was chosen using an LQR formulation with a state error weight of $Q_{LQR} = 1I$ and control weight of $R_{LQR} = 0.01I$, where I is an identity matrix of appropriate size for the relevant subsystem.

The control was then designed for each subsystem by solving the optimization problem (Equation 5.17) using the parameters shown in Table 5.1. The optimization problem was modeled using YALMIP [70] and solved using SDPT3 [71,72]. For better numerical stability, the problem was scaled to units of kilometers and hours.

The controller was designed with three parameters which were varied for each individual simulation. The upper bound on Q, q_{ub}, was chosen from 10 values in the range indicated, equally spaced on a log scale. Likewise, the optimization had an identical lower bound on Q, P, and R which was selected from 10 values in the given range, equally spaced on a log scale. Finally, the scaling between the P weight, p_w, and the R weight, r_w, was defined to be α and selected from 10 value in the indicated range, equally spaced on a linear scale.

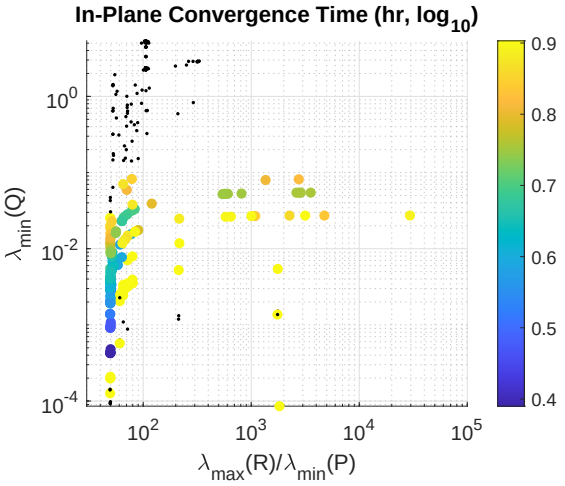

Fig. 5.1: Convergence time, shown on a $\log_{10}$ scale, for the in-plane dynamics. Simulations that did not converge within the simulation time are marked with a black dot.

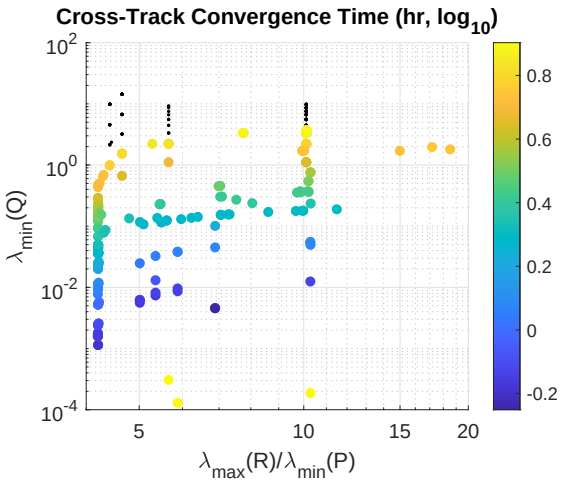

Fig. 5.2: Convergence time, shown on a $\log_{10}$ scale, for the cross-track dynamics. Simulations that did not converge within the simulation time are marked with a black dot.

The primary variable under consideration is the balance between the P and R weightings using the α parameter. For larger values of α, the objective function is weighted towards increasing the minimum eigenvalue of P while, for smaller values of α, the emphasis is on decreasing the maximum eigenvalue of R. Similarly, increasing the upper bound on Q allows for a larger design space and higher possible values of $\lambda_{min}(Q)$. Although not strictly necessary, the varying of the parameters for p_{lb}, r_{lb} and q_{ub} is also designed to explore a larger range of possible solutions. However, this results in some initial conditions being infeasible, due to the overlap between the upper and lower bounds on Q, and those are removed from consideration for the remainder of the analysis.

Each simulation begins with an initial condition

$$x_0 = \begin{bmatrix} 390\,\text{m} & 390\,\text{m} & 390\,\text{m} & 1.5\,\text{m/s} & 1.5\,\text{m/s} & 1.5\,\text{m/s} \end{bmatrix}^T \tag{5.18}$$

and the controller attempts to drive the system to the origin. Each simulation runs for 8 hrs with a control update step of 5 s. In total, 319 simulations were run.

In Figures 5.1 and 5.2, the convergence time is shown for each of the simulations as a function of $\lambda_{min}(Q)$ and $\lambda_{max}(R)/\lambda_{min}(P)$. The convergence time is defined as the time that

the subsystem converges to within 0.1 m of the origin. For some controllers, the convergence time exceeds the 8 hr simulation runtime and are marked with black dots.

From Equation 5.13, the upper bound on the convergence time decreases as the $\lambda_{max}(R)/\lambda_{min}(P)$ ratio decreases. For the in-plane subsystem (Figure 5.1), a decrease in the convergence time can be seen as the ratio decreases from a value of 100 to 14. However, for values of the ratio larger than 100, there does not appear to be any change in the convergence time as the ratio varies. For the cross-track subsystem (Figure 5.2), the ratio does not seem to have a practical effect on the convergence time.

However, for both subsystems, the convergence time tends to decrease with the $\lambda_{min}(Q)$ value. This is explained by the fact that the convergence time is a function of T_0 which, in turn, is a function of Q. Overall, the choice of Q to determine T_0 seems to have a stronger impact on the overall convergence time, relative to the R to P ratio. However, for very small $\lambda_{min}(Q)$, around 10^{-4}, the convergence times appear to increase again, showing that the trend is generally applicable but not guaranteed.

For comparison, the upper convergence time bound is shown in Figures 5.3 and 5.4 for the in-plane and cross-track subsystems, respectively. Here, the decrease in the upper bound with the decreasing R to P ratio is seen as well. Additionally, the decrease in the convergence bound with decreasing $\lambda_{min}(Q)$ is also seen. However, the upper bound is significantly higher than the observed times, showing that the bound is a very conservative estimate of the convergence time.

In Figures 5.5 and 5.6, the maximum observed control for the in-plane and cross-track subsystems are shown, respectively. The points with a red border indicate control magnitudes which exceed the theoretical maximum. As predicted by Equation 5.16, as $\lambda_{min}(Q)$ increases, the instantaneous control of each subsystem tends to decrease. Since the controller only acts on the Brunovsky dynamics, the Brunovsky transformation introduces an additional control which is unaccounted for. For many of the simulations, the control bound is sufficiently large, and the Brunvosky control sufficiently small, that the addition is not enough to exceed

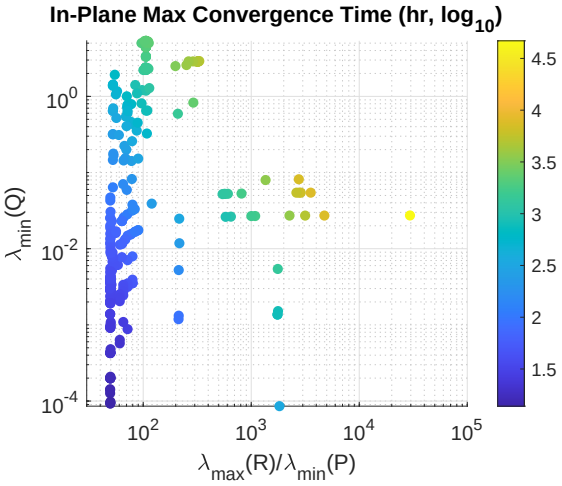

Fig. 5.3: Upper convergence time bound, shown on a $\log_{10}$ scale, for the in-plane dynamics.

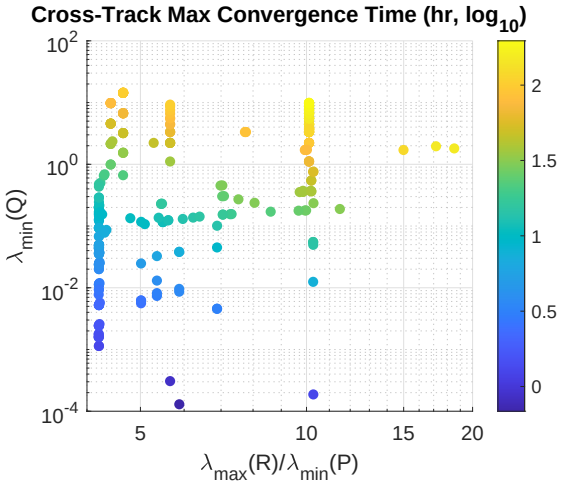

Fig. 5.4: Upper convergence time bound, shown on a $\log_{10}$ scale, for the cross-track dynamics.

the expected bounds. However, for tighter control bounds, the additional control is enough to exceed the limit.

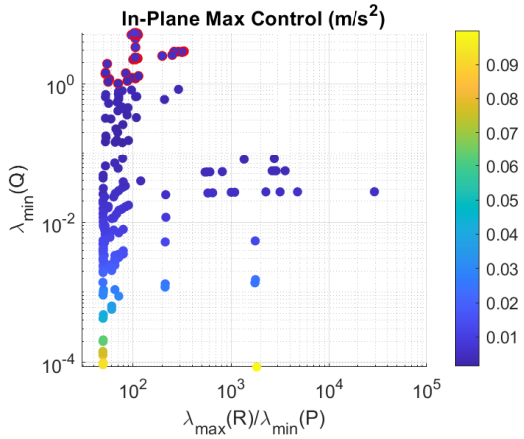

Fig. 5.5: Maximum control for the in-plane dynamics. Points outlined in red exceed the theoretical bounds.

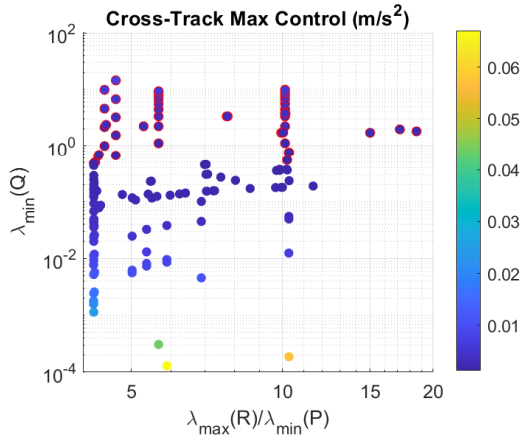

Fig. 5.6: Maximum control for the cross-track dynamics. Points outlined in red exceed the theoretical bounds.

For comparison, the theoretical limits are shown in Figures 5.7 and 5.8 for the in-plane and cross-track dynamics, respectively, which show the expected correlation between $\lambda_{min}(Q)$ and the control bound. Additionally, these bounds are less conservative than the

convergence time limits. The difference between the convergence time bound and the observed times differed by two or more orders of magnitude. However, the control bounds and observed controls only differ by a factor of 2 to 5.

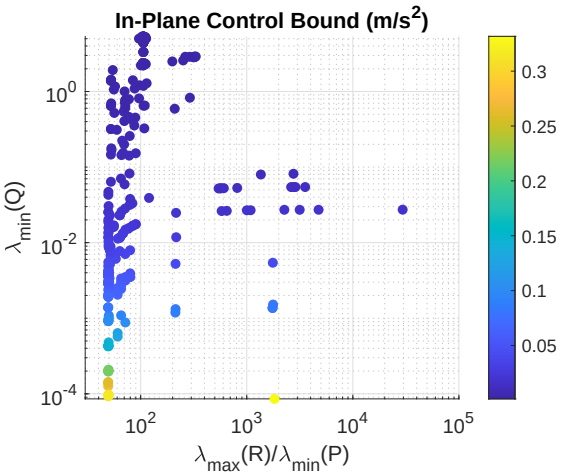

Fig. 5.7: Maximum control bound for the in-plane, CW dynamics.

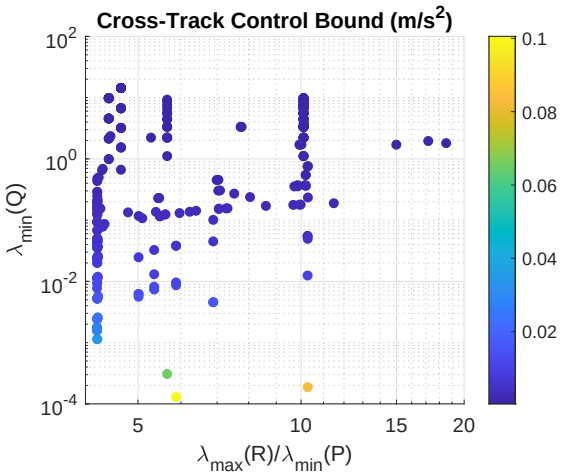

Fig. 5.8: Maximum control bound for the cross-track, CW dynamics.

5.2.2 Unperturbed CW Dynamics

To verify the performance of the controller and the usefulness of the optimization approach, a series of simulations were also run using the unperturbed CW dynamics. A single pair of common Lyapunov controllers was used for each simulations with one controlling the single-input in-plane subsystem and one controlling the cross-track subsystem. The controller was developed by selecting bounds on Q, P, and R that generally corresponded to low control usage and fast convergence time. These bounds were then used in the optimization approach to generate the control matrices, shown in Table 5.2. The same stabilizing feedback gain K which was used in the preceding analysis was also used for this controller.

A total of 50 simulations were run with the initial states randomly selected from those that satisfied

$$|x_0| = \begin{bmatrix} 390\,\text{m} & 390\,\text{m} & 390\,\text{m} & 1.5\,\text{m/s} & 1.5\,\text{m/s} & 1.5\,\text{m/s} \end{bmatrix}^T \tag{5.19}$$

Table 5.2: Matrices and associated eigenvalues of the controller for the in-plane (IP) and cross-track (CT) subsystems.

Parameter	Matrix	Eigenvalues
Q_{IP}	$\begin{bmatrix} 0.3007 & 0.3094 & 0.1368 & 0.0061 \\ 0.3094 & 0.6249 & 0.3303 & 0.0148 \\ 0.1368 & 0.3303 & 0.3303 & 0.0148 \\ 0.0061 & 0.0148 & 0.0148 & 0.0061 \end{bmatrix}$	$\begin{bmatrix} 0.0055 \\ 0.0782 \\ 0.1785 \\ 1.0000 \end{bmatrix}$
P_{IP}	$\begin{bmatrix} 0.1221 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.1221 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.1221 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.1221 \end{bmatrix}$	$\begin{bmatrix} 0.1221 \\ 0.1221 \\ 0.1221 \\ 0.1221 \end{bmatrix}$
R_{IP}	$\begin{bmatrix} 2.4060 & 2.1656 & 0.8207 & 0.0305 \\ 2.1656 & 3.7494 & 1.6514 & 0.0591 \\ 0.8207 & 1.6514 & 1.3213 & 0.0444 \\ 0.0305 & 0.0591 & 0.0444 & 0.0123 \end{bmatrix}$	$\begin{bmatrix} 0.0108 \\ 0.4038 \\ 1.0272 \\ 6.0472 \end{bmatrix}$
Q_{CT}	$\begin{bmatrix} 0.9977 & 0.0475 \\ 0.0475 & 0.0477 \end{bmatrix}$	$\begin{bmatrix} 0.0453 \\ 1.0001 \end{bmatrix}$
P_{CT}	$\begin{bmatrix} 0.9504 & 0.0000 \\ 0.0000 & 0.9504 \end{bmatrix}$	$\begin{bmatrix} 0.9504 \\ 0.9504 \end{bmatrix}$
R_{CT}	$\begin{bmatrix} 3.9910 & 0.1426 \\ 0.1426 & 0.0954 \end{bmatrix}$	$\begin{bmatrix} 0.0902 \\ 3.9962 \end{bmatrix}$

Each simulation was run for 6 hrs with a 0.5 s update step.

In Figures 5.9 and 5.10, the position errors for the in-plane and cross-track subsystems are shown, respectively. For the in-plane errors, the trajectory is generally smooth and all simulations end with approximately the same magnitude of error. However, over the course of the simulations, there are a wide range of position errors seen. Some simulations never exceed the initial the initial error magnitude while others experience errors as high as 1.5×10^4 m. In contrast, some cross-track errors show a small increase in the error, but this never exceeds 800 m. Additionally, the cross-track errors converge faster, likely a result of the smaller number of dimensions in this subsystem.

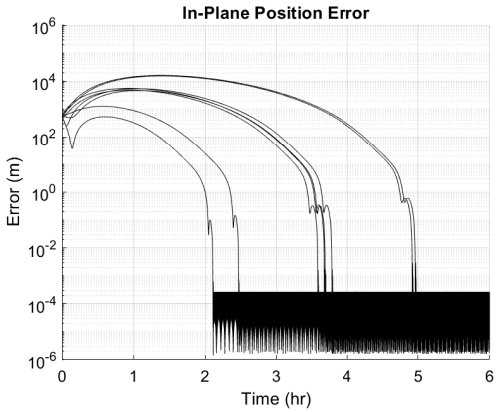

Fig. 5.9: In-plane position error for the unperturbed CW dynamics.

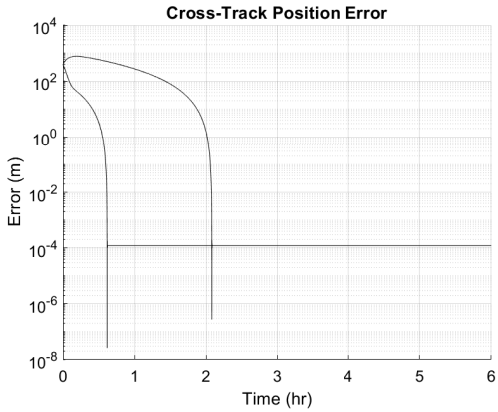

Fig. 5.10: Cross-track position error for the unperturbed CW dynamics.

In Figures 5.11 and 5.12, the control plots for the in-plane and cross-track subsystems, respectively, are shown. In both figures, the dashed line indicates the theoretical maximum bound on the control for that subsystem, based on the K feedback gain and Q matrix used in the controller. In all cases, the controls are less than this bound. The bound on the cross-track controls appears to be more conservative than its in-plane counterpart with the difference between the maximum control and the bound being about an order of magnitude. The in-plane control shows a gap of about $0.03\,\mathrm{m/s^2}$.

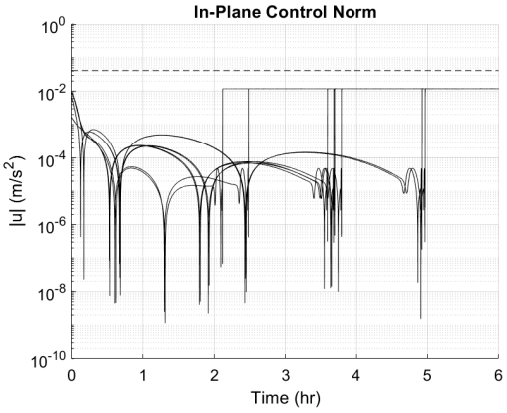

Fig. 5.11: In-plane control usage for the unperturbed CW dynamics. The red line indicates the theoretical maximum control bound.

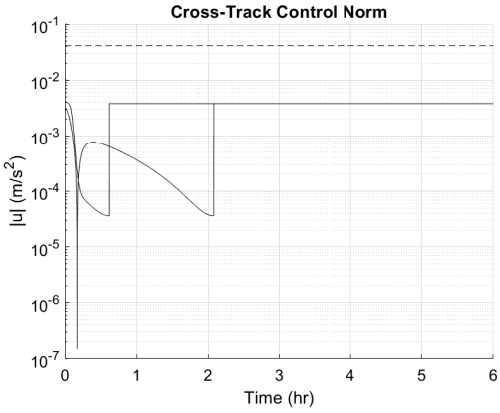

Fig. 5.12: Cross-track position error for the unperturbed CW dynamics. The red line indicates the theoretical maximum control bound.

Table 5.3: Physical parameters used for the nonlinear spacecraft simulation.

Parameter	Value
Initial Wet Mass	$24\,\mathrm{kg}$
I_{SP}	$160\,\mathrm{s}$
Projected Surface Area	$0.048\,\mathrm{m}^2$
C_D	2.2
Reference Orbit	$a = 6\,878\,000\,\mathrm{m}$ $e = 1 \times 10^{-4}$ $i = 170^\circ$ $\Omega = 126^\circ$ $\omega = 0^\circ$ $\nu = 0^\circ$

Both subsystems show a high degree of chatter at the end of the simulations. Since the direction, but not the magnitude, is changing, the chatter appears as a straight line on these plots. Additionally, the magnitude of the chatter is equal to, or greater than, the maximum control seen over the nominal, non-chattering portions of the simulations. This is undesirable behavior since at this point the subsystems are using the most control to try and correct the smallest errors.

5.2.3 Two-Body Dynamics

A series of simulations were also run using the two-body dynamics with J2 and drag perturbations. A single controller was developed for each subsystem and each subsystem had a single input channel, resulting in unmatched disturbances along the X-axis. The same controller used with the CW dynamics (Table 5.2) was used with the two-body dynamics. The simulations were run using the parameters in Table 5.3. A total of 50 simulations were run with initial conditions that satisfied

$$|x_0| = \begin{bmatrix} 390\,\mathrm{m} & 390\,\mathrm{m} & 390\,\mathrm{m} & 1.5\,\mathrm{m/s} & 1.5\,\mathrm{m/s} & 1.5\,\mathrm{m/s} \end{bmatrix}^T \tag{5.20}$$

for the relative state of the spacecraft. Each simulation was run for 6 hrs with a 0.5 s update step.

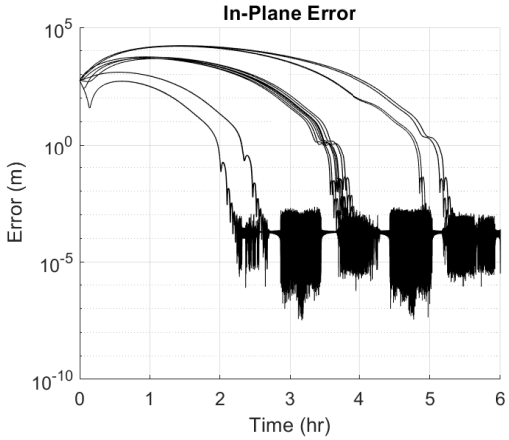

Fig. 5.13: Position error for the in-plane, two-body dynamics.

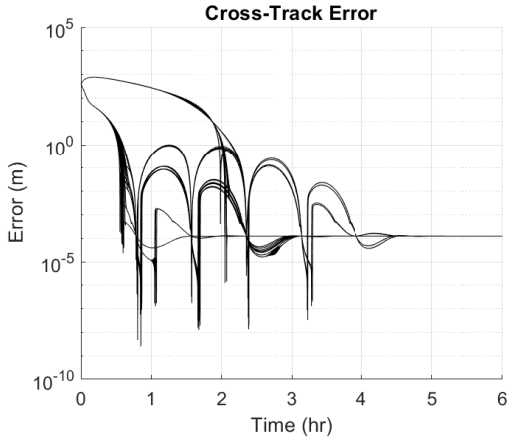

Fig. 5.14: Position error for the cross-track, two-body dynamics.

The position errors of the in-plane and cross-track subsystems of all 50 simulations are shown in Figures 5.13 and 5.14, respectively. Each simulation converges to a final, steady-state error within the 6 hr simulation duration. Calculating the theoretical upper bound of the convergence time for the in-plane subsystem gives a value of (34.9 ± 4.6) hrs while the cross-track subsystem has a theoretical bound of (5.1 ± 0.3) hrs. Similar to the CW results, the upper bound of the in-plane subsystem is very conservative and not a good estimate of the actual convergence time. However, the cross-track convergence time is less conservative and produces a bound which can be used to estimate the actual convergence time. Interestingly, the in-plane controller is able to stabilize the system about the origin despite the unmatched distrubance along the X-axis.

In Figures 5.15 and 5.16, the control magnitudes for the in-plane and cross-track subsystems are shown, respectively. The red dashed lines in each plot indicates the maximum theoretical control from the common Lyapunov approach for each subsystem. Due to the additional control from the Brunovsky transformation, the bound for the common Lyapunov approach is not guaranteed to be maintained for the full system. However, in all simulations the total control stayed below the theoretical bound.

The bounds for both subsystems appear to be reasonable approximations of the actual maximum control. For the in-plane control, the bound is found to be $0.0416\,\mathrm{m/s^2}$ with a

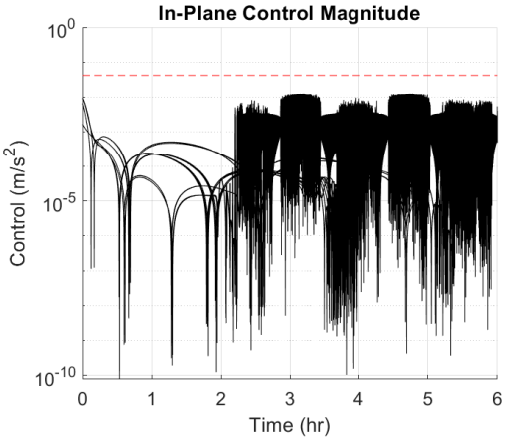

Fig. 5.15: Control usage for the in-plane, two-body dynamics. The dashed red line indicates the maximum theoretical control.

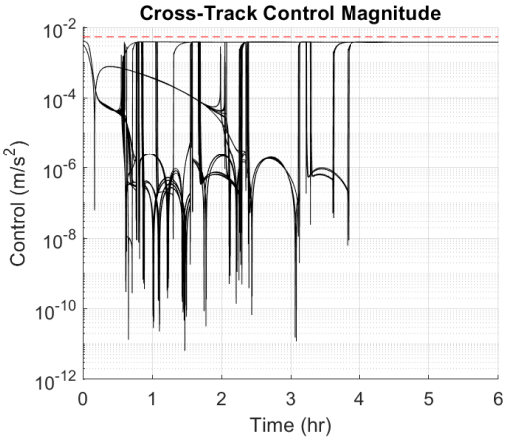

Fig. 5.16: Control usage for the cross-track, two-body dynamics. The dashed red line indicates the maximum theoretical control.

maximum actual control of $0.0122\,\mathrm{m/s^2}$. Similarly, the cross-track control is bounded by $0.0054\,\mathrm{m/s^2}$ with an actual maximum control of $0.0042\,\mathrm{m/s^2}$.

One item of note is that both subsystems exhibit chatter in the controls. For the in-plane subsystem, the chatter is apparent as the control magnitudes rapidly oscillate. However, for the cross-track subsystem, the control direction is rapidly changing but maintaining constant magnitude and appears in the figure as a constant, non-zero control magnitude. However, in all cases, the theoretical bound is maintained on the subsystem controls.

5.3 Conclusion

This chapter implemented a common Lyapunov control which guarantees finite time convergence. The effect of the three matrices used in the control are examined and the theoretical bounds on the convergence time and maximum control were computed. A semidefinite program was used to find control laws which minimize the convergence time and control usage. When tested using both the CW and two-body dynamics, the control laws were found to suitably bound the maximum control but were overly conservative with regards to the convergence time.

CHAPTER 6

PREDEFINED-TIME CONTROL VIA BACKSTEPPING

While the convergence time of finite-time or fixed-time control laws can be quantified to some extent, the connection between the control law and the convergence time is generally non-obvious. This poses a challenge when attempting to design a control law with specific convergence time guarantees. As a result, another class of controllers, known as predefined-time controllers, have been developed which use the desired convergence time as a parameter.

This chapter uses the Brunovsky transformation of the relative orbital dynamics with a predefined-time backstepping controller to guarantee predefined-time convergence to a desired relative trajectory. The system is tested in both perturbed and unperturbed situations with modifications made to the single-input Brunovsky transformation to guarantee convergence even when affected by perturbations.

6.1 Predefined-Time Control

A predefined-time control is one where the convergence time bound is explicitly used as a parameter in the control law. Consider the time-varying system

$$\dot{x} = f(t, x, T_c), \quad \forall t \geq 0 \tag{6.1}$$

with state $x \in \mathbb{R}^n$, convergence time bound $T_c \in \mathbb{R}_{>0}$, and $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ which is piecewise continuous in t and locally Lipschitz in x. Furthermore, let $x(t, x_0, T_c)$ represent the solution of Eq. 6.1 at time t with initial condition x_0 [3].

We can now define predefined-time stability.

Definition 6.1.1 ([3, Definition 1])**.** The origin is *predefined-time stable* if $\forall t \geq T_c$, $x(t, x_0, T_c) = 0$.

Note the difference in the definition of predefined-time stability compared to finite-time or fixed-time stability. While finite-time or fixed-time stability guarantee that the system will go to zero within some finite amount of time, the actual convergence bound is either unknown, in the case of finite-time stability, or non-trivially related to the tuning parameters in the control law, in the case of fixed-time stability. In contrast, predefined-time stability provides a control parameter with a known relationship to the upper-bound on the convergence time.

As with other control laws, there are Lyapunov conditions which, if met, guarantee predefined-time convergence. These conditions are presented in the following theorem.

Theorem 11 ([39, Corollary 12])**.** If there exists a continuous radially unbounded function $V : \mathbb{R}^n \to \mathbb{R}$ such that

$$\begin{aligned} V(x) &\geq 0 \\ V(x) = 0 &\Leftrightarrow x = 0 \end{aligned} \tag{6.2}$$

and any solution $x(t)$ to $\dot{x} = f(x)$ satisfies

$$\dot{V}(x) \leq -\frac{1}{pT_c}\exp(V(x)^p)V(x)^{1-p}, \quad \text{for } x \in \mathbb{R}^n \setminus \{0\} \tag{6.3}$$

with $T_c \in \mathbb{R}_{>0}$ and $0 < p \leq 1$, then the origin is predefined-time stable with predefined time T_c.

As with other convergence guarantees, finding a suitable Lyapunov function and control law for a system can be difficult if starting from scratch. However, various approaches have been developed which provide control laws for specific dynamics. In [73–76], the controller design is approached through direct application of the Lyapunov conditions. This is done for systems where each state has a control input [73, 75], a double integrator with matched disturbances [74], and single-input, single-output systems [76]. Another approach was developed in [77] for use with nonlinear, control-affine systems which uses optimal control techniques.

Neural-network techniques were used in [78, 79] for strict-feedback systems. In [78], they are used to approximate uncertain nonlinear dynamics in the system while in [79]

they are used to approximate parameters in the control laws. Similarly, in [4], an adaptive predefined-time backstepping control was introduced which was able to stabilize a nonlinear system which included some unknown set of parameters.

This chapter utilizes an approach presented in [3] referred to as predefined-time backstepping. This approach assumes a strict-feedback, single-input system with matched disturbances. While the relative orbital dynamics are not an immediate match for these assumptions, a few changes can be made which allow for the transformation into an appropriate form. First, the relative orbital dynamics can be divided into in-plane and out-of-plane dynamics that act independently of each other. Then, since the in-plane dynamics are completely controllable using only the Y-axis control, a single-input form of chained integrators can be constructed using the Brunovsky form and satisfy the required form while the out-of-plane dynamics naturally match the form needed by predefined-time backstepping.

As will be shown, this approach works well for the unperturbed system. However, since perturbations can affect all axes, the controller fails to properly converge when perturbations are present. This is addressed by the use of a multi-input Brunovsky form which allows for perturbation rejection along all axes.

6.2 Predefined-Time Backstepping Control

The work in [3] presents a predefined-time control for an nth order system of the form

$$\dot{x}_i = f_i(\bar{x}_i) + x_{i+1}, \quad i = 1, \ldots, n-1 \tag{6.4}$$

$$\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u + d(t) \tag{6.5}$$

where $\bar{x}_i$ is the vector of all states up to and including i ($[x_1, \ldots, x_i]$), f_i and f_n are known smooth nonlinear functions, g_n is a known continuous input function, $u \in \mathbb{R}$ is the system input, and $d(t)$ is an unknown, but bounded, disturbance such that for all t, $|d(t)| < D$ with $D \in \mathbb{R}_{\geq 0}$.

While the CW dynamics do not match the form needed for the predefined-time backstepping control, the Brunovsky forms (Section 3.3) do match. This allows for the out-of-plane

dynamics to be controlled independently from the in-plane dynamics. Additionally, the controller for the in-plane dynamics can be designed assuming controls along the X- and Y-axes or only along the Y-axis. If control is assumed along the X- and Y-axes, then two separate predefined-time backstepping controllers are developed for each independent, in-plane Brunovsky subsystem while only one controller is needed in the case of the single-input control.

As mentioned in Section 3.3, the C matrix resulting from the Brunovsky transformation generally exists outside any control law using that form. Restating the Brunovsky transformation,

$$A_B = T_D^{-1}(A + BC)T_D \tag{6.6}$$

$$B_B = T_D^{-1}B \tag{6.7}$$

$$z = T_D^{-1}x \tag{6.8}$$

$$v = u - Cz \tag{6.9}$$

$$\dot{z} = A_B z + B_B v \tag{6.10}$$

which shows that when a controller calculates v using the Brunovsky dynamics, the matrix C is only introduced when the actual system control u is calculated. However, a slight rearranging of terms in the Brunovsky from allows for full integration of the system into the predefined-time backstepping control for the single input in-plane dynamics. If we combined Equations 6.8, 6.9, 6.10 then we get

$$\dot{z} = A_B z + B_B u - B_B C T_D z. \tag{6.11}$$

Now we examine the dynamics for the n^{th} and final state in z. From the definition of the Brunovsky form, the last row of A_B is comprised of zeros so $(A_B z)_n = 0$ where $(A_B z)_n$ represents the n^{th} element of $A_B z$. Similarly, the last element of B_B is defined to be 1 so

$(B_B u)_n = u$ and $(B_B C T_D z)_n = C T_D z$. Therefore,

$$\dot{z}_n = u - C T_D z \tag{6.12}$$

and defining $f_n(z) = -C T_D z$ results in the Brunovsky form matching the desired predefined-time backstepping control form.

Note that the Brunovsky dynamics are still in terms of z, the transformed coordinates, but the control is now expressed in terms of u, the control applied to the nominal system. This means that the actual states must be converted to the Brunovsky states before calculating the control but that once the control is determined, the feedback gain does not need to be applied before it is inputted to the system.

This approach, which uses the feedback term within the control law, only works with the out-of-plane and single-input in-plane dynamics. With the dual-input in-plane dynamics, the appearance of the $\dot{y}$ and $\dot{x}$ terms in the x and y dynamics, respectively, fail to satisfy the requirements placed on the nonlinear dynamics $f_i(x_i)$.

6.2.1 Tuning Function

The controller requires a smooth time-varying tuning function $\rho(t)$ which satisfies three conditions. First, $\rho(t)$ and its derivatives $\rho^{(i)}(t), (i = 1, \ldots, n)$, where n is the number of states in the system, are bounded and continuous for $t \geq 0$. Second, $\rho(t) = 0$ for all $t \geq T_c$. And third, $\rho(0)$ and $\rho^{(j)}(0),\ (j = 1, \ldots, n-1)$ satisfy

$$\rho(0) = x_1(0) - y_d(0) \tag{6.13}$$

$$\rho^{(j)}(0) = x_{j+1}(0) + \sum_{k=1}^{j} f_k^{(j-k)}(\bar{x}_j(0)) - y_d^{(j)}(0) \tag{6.14}$$

These requirements result in a total of $2n + 1$ constraints that must be satisfied. There are n constraints resulting from the initial conditions of $\rho(t)$ and its $n - 1$ derivatives, and there are $n + 1$ constraints at $t = T_c$ since $\rho(t)$ and all of its derivatives must be equal to zero.

The function ρ can be created by selecting a set of basis functions, which are not necessarily orthogonal, $\ell(k,x)$ with ρ defined as

$$\rho(t) = \begin{cases} \sum_{k=0}^{2n} a_k \ell(k,t), & t < T_c \\ 0, & t \geq T_c \end{cases}. \tag{6.15}$$

The coefficients a_k can then be found as

$$a = L^{-1}M \tag{6.16}$$

Here L is a matrix of all the basis functions and their derivatives, evaluated at both $t = 0$ and $t = T_c$, and expressed as

$$L = \begin{bmatrix} \ell(0,0) & \ell(1,0) & \dots & \ell(2n,0) \\ \dot{\ell}(0,0) & \dot{\ell}(1,0) & \dots & \dot{\ell}(2n,0) \\ \vdots & \vdots & \ddots & \vdots \\ \ell^{(n)}(0,0) & \ell^{(n)}(1,0) & \dots & \ell^{(n)}(2n,0) \\ \ell(0,T_c) & \ell(1,T_c) & \dots & \ell(2n,T_c) \\ \dot{\ell}(0,T_c) & \dot{\ell}(1,T_c) & \dots & \dot{\ell}(2n,T_c) \\ \vdots & \vdots & \ddots & \vdots \\ \ell^{(n+1)}(0,T_c) & \ell^{(n+1)}(1,T_c) & \dots & \ell^{(n+1)}(2n,T_c) \end{bmatrix} \tag{6.17}$$

The M vector consists of the constraints on the initial and terminal conditions of ρ and its derivatives, expressed as

$$M = \begin{bmatrix} \rho(0) & \dot{\rho}(0) & \dots & \rho^{(n)}(0) & 0_{1\times 2n+1} \end{bmatrix}^T \tag{6.18}$$

6.2.2 Controller Formulation

The control law is defined as

$$u = g_n^{-1}(x_n)\left[-f_n(\bar{x}_n) + \dot{\alpha}_n - D\,\mathrm{sign}(e_n) - e_{n-1}\right] \tag{6.19}$$

where $\dot{\alpha}_n$ is the time derivative of the nth virtual control law α, e_n are error states, and D is a constant chosen such that $|d(t)| < D$ for $t \geq 0$. The virtual control laws and error states are defined as follows. The error states e are defined to be

$$e_1 = x_1 - y_d(t) - \rho(t) \tag{6.20}$$

$$e_j = x_j - \alpha_j, \quad j = 2, \ldots, n \tag{6.21}$$

where $y_d(t)$ is the desired trajectory and $\rho(t)$ is a tuning function which will be defined later. The desired trajectory $y_d(t)$ and its n time derivatives should be piecewise continuous and bounded on $t \geq 0$. The virtual control laws α are defined as

$$\alpha_2 = -f_1(x_1) + \dot{y}_d(t) + \dot{\rho}(t) \tag{6.22}$$

$$\alpha_{j+1} = -f_j(x_j) + \dot{\alpha}_j - e_{j-1}. \tag{6.23}$$

The derivatives of the error states and virtual control laws are

$$\dot{e}_1 = e_2 \tag{6.24}$$

$$\dot{e}_j = e_{j+1} - e_{j-1}, \quad j = 2, \ldots, n-1 \tag{6.25}$$

$$\dot{e}_n = d(t) - D\mathrm{sign}(e_n) - e_{n-1} \tag{6.26}$$

$$\dot{\alpha}_2 = -f_1^{(1)}(x_2) + \ddot{y}(t) + \ddot{\rho}(t) \tag{6.27}$$

$$\begin{aligned}\dot{\alpha}_{j+1} = &-\sum_{i=1}^{j} f_i^{j+1-i}(\bar{x}_{j+1}) + y_d^{(j+1)}(t) \\ &+ \rho^{j+1}(t) - \sum_{k+1}^{j-1} e_k^{(j-k)}, \quad j = 2, \ldots, n-1\end{aligned} \tag{6.28}$$

Although $\dot{e}_n$ is defined as a function of the disturbance $d(t)$, it is only used in the stability proof and not in the control law itself.

Once the control law is developed, the control can be calculated at each time t given the current time and state and then applied to the system.

6.2.3 Tuning Function Optimization

The tuning function, as formulated above, has $2n+1$ constraints and $2n+1$ variables which only allows for a single solution. However, by adding additional basis functions to ρ, additional flexibility can be gained to influence the transient response.

Examining the control law shows that the control response is related to $\rho^{(n+1)}$ and so the characteristics of ρ can be used to influence the control. Since $\rho^{(n+1)}$ only comprises a portion of the total control, it is not possible to use it to constrain the control profile to satisfy specific criteria such as staying within specified bounds. However, general control trends can be influenced such as minimizing the L_∞ norm of ρ to limit instantaneous maximum control.

6.2.4 Stability Analysis

The stability analysis presented is drawn directly from [3]. Assuming that a proper ρ has been determined we can now consider the initial conditions. From the definition of ρ we know that

$$\begin{aligned} \rho(0) &= x_1(0) - y_d(0) \\ \rho^{(j)}(0) &= x_{j+1}(0) + \sum_{k=1}^{j} f_k^{(j-k)}(\bar{x}_j(0)) - y_d^{(j)}(0) \quad j = 1, \ldots, n-1 \end{aligned} \tag{6.29}$$

The initial conditions for the virtual control laws are

$$\begin{aligned} \alpha_2(0) &= -f_1(x_1(0)) + \dot{y}_d(0) + \dot{\rho}(0) \\ \alpha_3(0) &= -f_2(\bar{x}_2(0)) - f_1^{(1)}(\bar{x}_2(0)) + \ddot{y}_d(0) + \ddot{\rho}(0) - e_1(0) \\ \alpha_{j+1}(0) &= -\sum_{i=1}^{j} f_i^{j-i}(\bar{x}_j(0)) + y_d^{(j)}(0) + \rho^{(j)}(0) - \sum_{k=1}^{j-2} e_k^{(j-1-k)}(0) - e_{j-1}(0), \quad j = 3, \ldots, n-1 \end{aligned} \tag{6.30}$$

Combining the initial conditions for ρ and the virtual control laws gives

$$\begin{aligned} \alpha_2(0) &= x_2(0) \\ \alpha_3(0) &= x_3(0) - e_1(0) \\ \alpha_{j+1}(0) &= x_{j+1}(0) - \sum_{k=1}^{j-2} e_k^{(j-1-k)}(0) - e_{j-1}(0), \quad j = 3, \ldots, n-1 \end{aligned} \tag{6.31}$$

For the initial e_1 error,

$$\begin{aligned} e_1(0) &= x_1(0) - y_d(0) - \rho(0) \\ &= x_1(0) - y_d(0) - x_1(0) - y_d(0) \\ &= 0 \end{aligned} \tag{6.32}$$

where the initial condition for $\rho(0)$ has been applied. For the initial e_2 value

$$\begin{aligned} e_2(0) &= x_2(0) - \alpha_2(0) \\ &= x_2(0) - x_2(0) \\ &= 0 \end{aligned} \tag{6.33}$$

where we have used the initial α_2 value determined from Eq. 6.31. Similarly,

$$\begin{aligned} e_3(0) &= x_3(0) - \alpha_3(0) \\ &= x_3(0) - x_3(0) - e_1(0) \\ &= e_1(0) \\ &= 0 \end{aligned} \tag{6.34}$$

We now turn our attention to the error derivatives that appear in $\alpha_{j+1}(0)$ (Eq. 6.31). Note that expanding $e_a^{(b)}$ includes the error terms e_1 through e_{a+b} but does not include any higher terms ($e_{a+b+1}, \ldots, e_n$). Thus, the $\sum_{k=1}^{j-2} e_k^{(j-1-k)}$ term, for $j = 3, \ldots, n-1$, contains the e_1, ..., e_{j-1} terms but not the e_j, ..., e_n terms. In the specific case of $j = 3$, we have already determined that $e_1(0) = e_2(0) = 0$ and so $\sum_{k=1}^{j-2} e_k^{(j-1-k)}(0) = 0$ since it is only

comprised of $e_1(0)$ and $e_2(0)$. This gives,

$$\begin{aligned} e_4(0) &= x_4(0) - \alpha_4(0) \\ &= x_4(0) - x_4(0) - \sum_{k=1}^{1} e_k^{(2-k)}(0) - e_2(0) \\ &= -e_2 \\ &= 0 \end{aligned} \tag{6.35}$$

This process can be continued for all error values to show that $e_i(0) = 0$ for $i = 1, \ldots, n$.

We now define a Lyapunov function to be

$$V = \frac{1}{2} \sum_{i=1}^{n} e_i^2 \tag{6.36}$$

with derivative

$$\dot{V} = e_n d(t) - e_n D \text{sign}(e_n) \leq 0 \tag{6.37}$$

which follows from Equations 6.24, 6.25, and 6.26. With $e_1(0) = e_2(0) = \ldots = e_n(0) = 0$, then $V(0) = 0$ and since $\dot{V}(t) \leq 0$ for $t \geq 0$, $V(t) = 0$. Furthermore, since $V(T_c) = 0$ implies $e_1(T_c) = 0$, and by definition $\rho(T_c) = 0$, then $x_1(T_c) = y_d(T_c)$, satisfying the predefined-time requirement that the system converges to the desired trajectory by time T_c.

However, this analysis shows a nuance in the predefined-time definition used in [3], and a similar one in [75]. To restate, the definition says that given a system $\dot{x} = f(x, t, T_c)$ for all $t \geq 0$ with $x \in \mathbb{R}^n$, $T_c \in \mathbb{R}_{>0}$, $t \in \mathbb{R}_{\geq 0}$, and $f : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{>0}$ is piecewise continuous in t and locally Lipschitz in x, it is predefined-time stable if for all $x_0 \in \mathbb{R}^n$ and all $t \geq T_c$, $x(t, x_0, T_c) = 0$. From the stability analysis, this was shown to be true for this control approach. However, this presumes a nominal, unperturbed system since, given random perturbations, one cannot guarantee that the system will be strictly 0 for all $t \geq T_c$. The same ambiguity exists with finite-time and fixed-time stability definitions since the definitions assume a constant state of 0 after some specific time.

The Lyapunov condition given in Theorem 11 allows for the inclusion of disturbances into the finite-time analysis. However, as noted in [75], this can result in control profiles that may be impossible to implement due to the control's high frequency or high magnitude response. This has led to bounded predefined-time controllers which guaranteed predefined-time convergence to the vicinity the origin [75]. Ultimately, the controller used here takes a middle ground approach where the nominal system is predefined-time stable but no bounding conditions are given, only that the control rejects perturbations in asymptotic time.

6.3 Controller Selection

The predefined-backstepping method was tested in a variety of simulations. First, it was applied to the system described in [3] to verify that the implementation is correct and matches the expected results. Second, the control was applied to a spacecraft differential drag rendezvous problem with no perturbations. Third, the control was applied to a full spacecraft formation flying problem with J2 and drag perturbations.

6.3.1 Example Application

The first test was to replicate the example from [3] to highlight the features of the controller. The system dynamics were defined as

$$\dot{x}_1 = \sin(x_1) + x_2 \tag{6.38}$$

$$\dot{x}_2 = 2\cos(x_1 x_2) + x_2^2 + x_3 \tag{6.39}$$

$$\dot{x}_3 = x_1 - x_3 - 2\sin(x_2 x_3) + (2 + \sin(x_3))u + d(t) \tag{6.40}$$

where $d(t) = \sin(2t)$ and the reference signal is $y_d(t) = \sin(t) + \sin(0.5t)$.

The initial conditions were set as $x_0 = \begin{bmatrix} -1 & -1 & 1 \end{bmatrix}^T$ with the disturbance bounded by $D = 1$. A polynomial tuning function was used of the form

$$\rho(t) = \begin{cases} \sum_{k=0}^{2n} a_k t^k, & t < T_c \\ 0, & t \geq T_c \end{cases} \tag{6.41}$$

with the number of states $n = 3$.

The simulation was run from $t = [0,\ 3]$ with the predefined time $T_c = 2$ and in Figure 6.1, the initial results can be seen with the trajectory, trajectory error, and control usage seen in the three plots. As desired, the system converges to the desired trajectory shortly before the $t = 2$ mark and follows it afterwards. Of particular note is the fact that the controller is able to continue to follow the trajectory and reject perturbations even after the $T = 2$ convergence time. This shows that the controller only needs to be designed and implemented a single time which converges by the designated time but continues to maintain the desired properties for all $t \geq T_c$.

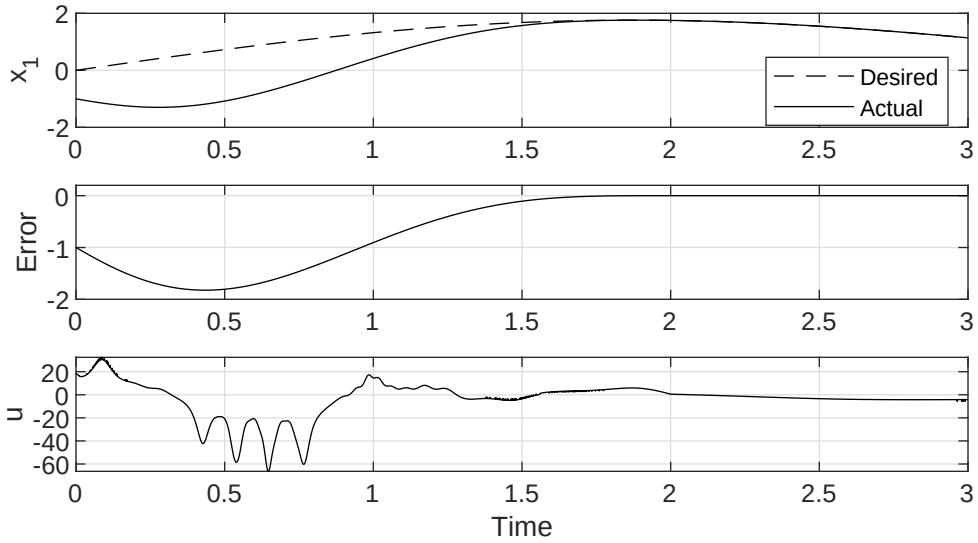

Fig. 6.1: Results matching showing the x_1 state, its error, and the control matching the paper results.

6.3.2 Differential Drag Rendezvous

The controller was tested on the in-plane differential drag model using the Brunovsky transformation (Section 3.3). The initial condition of the true dynamics was chosen to be $x_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$ and the dynamics were propagated according to the nominal in-plane CW dynamic model. Within the controller, the system state was transformed into the Brunovsky form and that was used to design the tuning function and calculate the transformed control v. Then the Brunovsky feedback control was added and the combined control was applied to the simulation.

A predefined-time of $T_c = 1$ was chosen for the initial sim and the results can be seen in Figure 6.2. As expected, all the states converge to 0 just before the predefined time is reached. Examining the control shows both the control v found using the backstepping method as well as the Brunovsky feedback term and the combination of the two. For this simulation, the backstepping control portion is seen to dominate the inherent Brunovsky feedback gain.

A second simulation was run with $T_c = 10$ (Figure 6.3) with significantly different results. While the states still smoothly converge before the predefined time, there is much more error in the states prior to that convergence. More importantly, the maximum control usage magnitude has increased by about 4 times, almost entirely due to the backstepping control.

This is a somewhat surprising result since, generally, for space dynamics an increase in the convergence time allows for a decrease in the control usage. Additionally, the previous simulation for $T_c = 1$ shows a result that would generally be more preferable in terms of a convergence time, control usage, and state deviation metric. This highlights the potentially double-edge sword of this approach since the controller consistently converges relatively close to the predefined time but does so at the cost of other potentially desirable considerations.

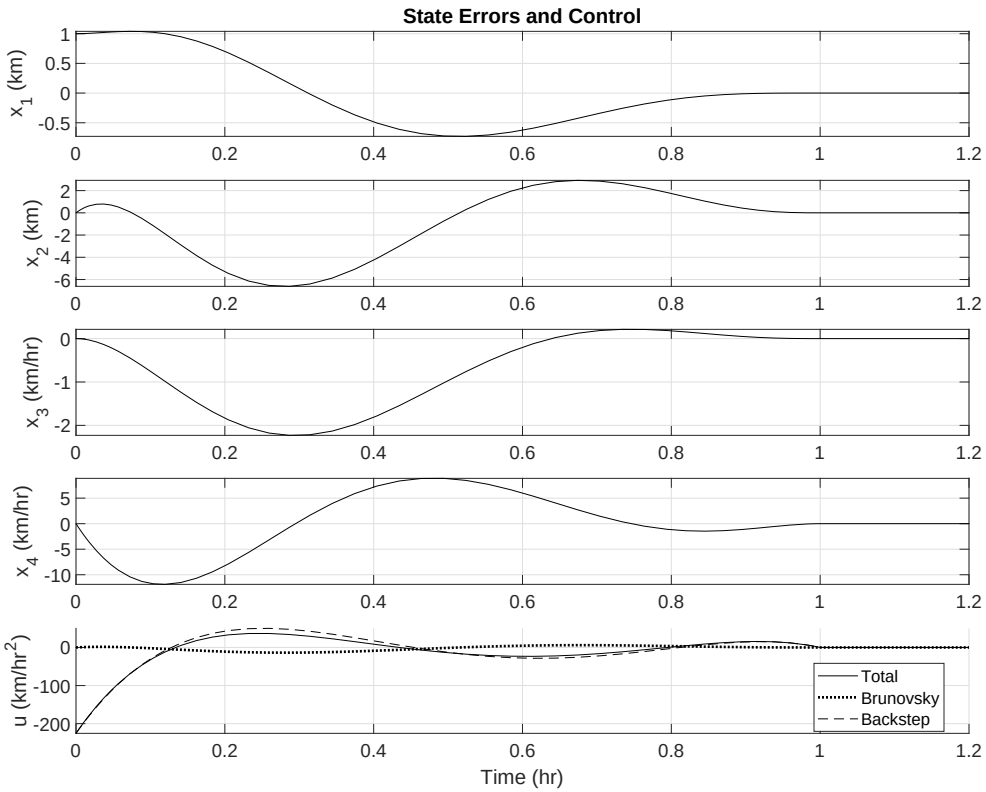

Fig. 6.2: Brunovsky results with $T_c = 1$.

6.3.3 ρ Optimization

The ability to optimize the tuning function ρ was tested by attempting to minimize the L_∞ norm of $\rho^{(n+1)}$. Since the control usage is proportional to $\rho^{(n+1)}$, this should, in theory, minimize the maximum control used to drive the system to the desired trajectory.

For this, a harmonic oscillator was used, defined as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 + u \end{aligned} \tag{6.42}$$

with initial condition $x = \begin{bmatrix} -1 & -2 \end{bmatrix}^T$ and a cutoff time of $T_c = 1$. The Legendre polynomials [80] are used as the basis functions. Note that although the Legendre polynomials

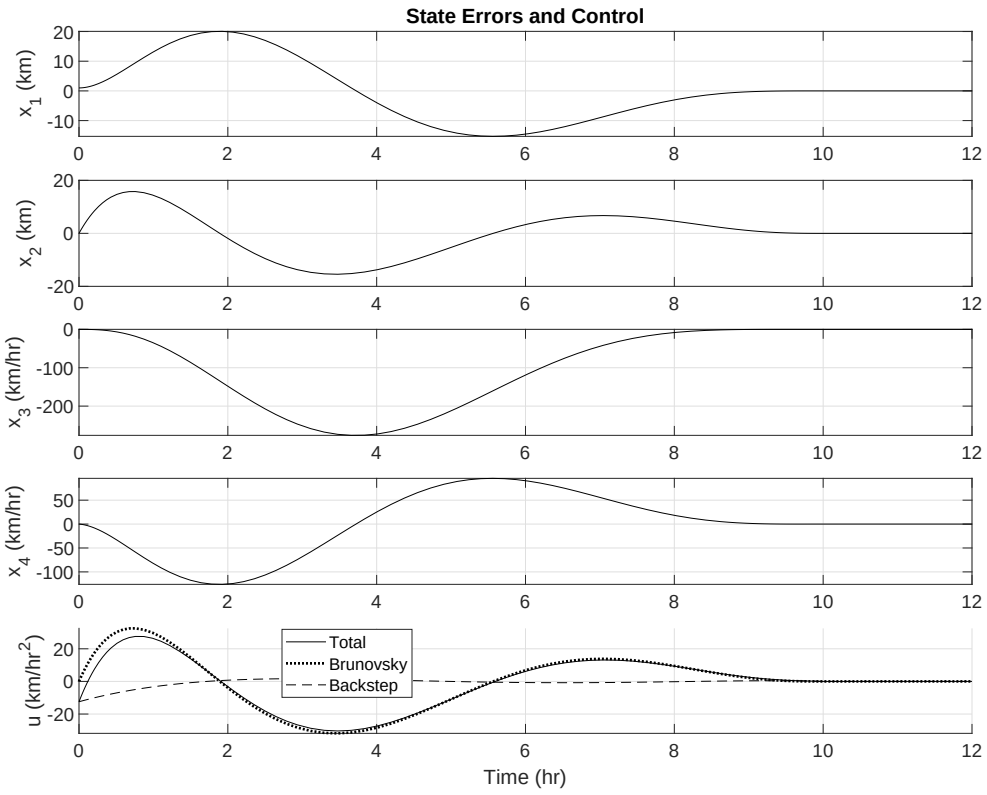

Fig. 6.3: Brunovsky results with $T_c = 10$.

are orthonormal on the interval $[0, 1]$, this is not a requirement for either the standard nor optimized approach to designing ρ.

The optimization is defined by solving for the coefficients a_k as

$$\begin{aligned} \min_{a_k, i=0,\ldots,2n+m} \quad & \max(|\rho^{(n+1)}(t)|) \\ \text{s.t.} \quad & La = M \\ & \rho(t) = \begin{cases} \sum_{k=0}^{2n+m} a_k \ell(k,t), & t < T_c \\ 0, & t \geq T_c \end{cases} \end{aligned} \tag{6.43}$$

where m is the number of extra basis functions and L and M are similar to their definitions in Equations 6.17 and 6.18, respectively, but with $2n + m + 1$ basis functions instead.

As implemented, optimization of ρ is done by sampling $\rho^{(n+1)}(t)$ at 1000 points between 0 and T_c and finding the maximum sampled magnitude. Although this naive method has weaknesses in its approach, notably the fact that the true maximum is likely to exist between the sampled points, it was deemed sufficient for the purposes of demonstrating the optimization of ρ.

In Figure 6.4, the default case without optimization is shown with a maximum value for $\ddot{\rho}(t)$ of 24. Adding three additional basis functions (Figure 6.5) allows the maximum magnitude to drop to 9.88. The actual control profile for the nominal and optimized controllers is shown in Figure 6.6. Note that while the control profiles mimic their respective $\ddot{\rho}$ functions, the actual values differ due to the inclusion of the other elements in the control law which, for this system, is due to the $f_2(\bar{x}_2) = -x_1$ term. However, the control profile and $\ddot{\rho}$ could also differ due to the inclusion of a non-zero desired trajectory or error corrections resulting from non-zero perturbations.

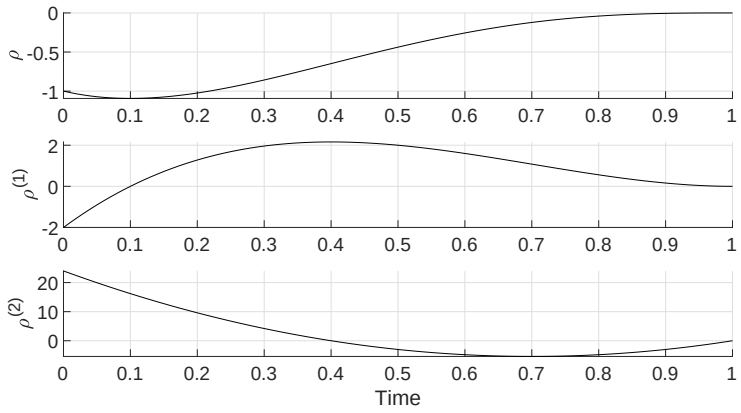

Fig. 6.4: $\rho(t)$ without any additional basis functions.

While this demonstrates the ability to influence the control via the tuning functions, there are limits. As noted, the primary limit is that the tuning function does not exactly correlate to the control profile. Additionally, while $\rho^{(n)}$ does not have an initial condition, it still needs to satisfy the conditions placed on $\rho^{(i)}$, $i = 0, \ldots, n-1$ which restricts the possible solutions.

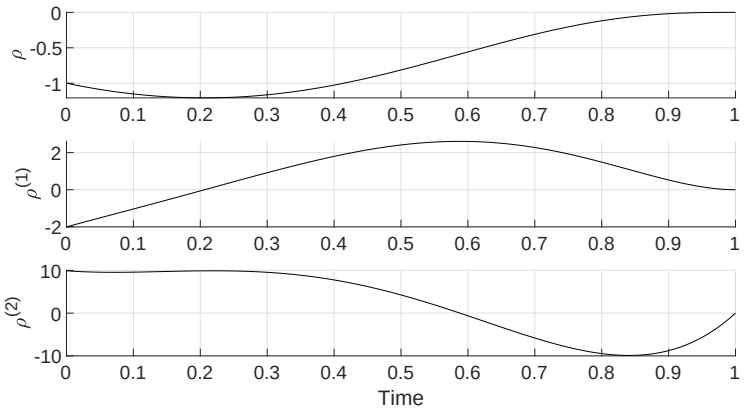

Fig. 6.5: $\rho(t)$ with three additional basis functions.

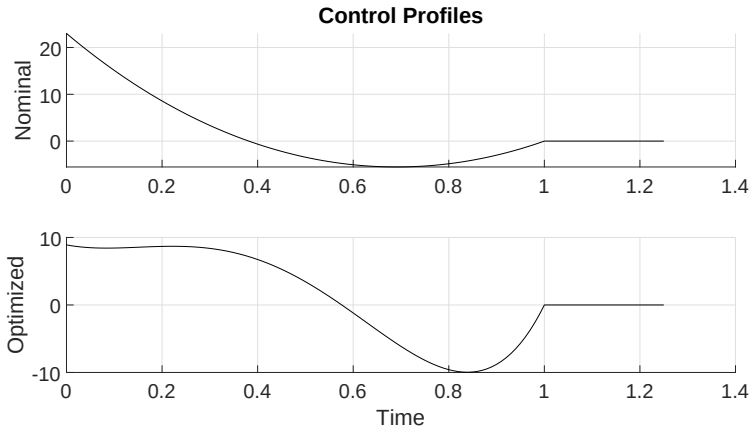

Fig. 6.6: Control profile for the nominal and optimized harmonic oscillator systems.

One variable that was not adjusted was the choice of basis functions. It is likely that a wise selection of basis functions could allow a more direct shaping of $\rho^{(n)}$ but this is left as future work.

6.4 Results

The finite-time backstepping method is finally applied to the spacecraft rendezvous scenario. A series of simulated environments were tested to determine the full limitations of the control law. First, the control law was tested using the CW dynamics for both the model and simulation with no added perturbations. Second, the control law was applied to the CW dynamics with a constant perturbation applied along the Y-axis to represent a drag

disturbance. Third, the controller was tested in a full 2-body simulation with actual J2 and drag effects. A total of 50 simulations are run for each scenario with the initial relative states selected such that

$$|x_0| = \begin{bmatrix} 390 & 390 & 390 & 1.5 & 1.5 & 1.5 \end{bmatrix}^T \tag{6.44}$$

The control inputs were updated every 10 s for each simulation.

6.4.1 Unperturbed CW Dynamics

For the unperturbed simulation, a spacecraft was initialized on one of the vertices of the allowable initial conditions for the guidance error and tasked with driving that error to zero using the single-input in-plane control. The terminal time for the controller is set to be 2.2 hrs and a perturbation bound of $7 \times 10^{-6}\,\mathrm{m/s^2}$ was used. No optimization of the tuning function was performed so the unperturbed dynamics could serve as a baseline reference.

In Figures 6.7 and 6.8, the LVLH error states can be seen for the in-plane and cross-track subsystems, respectively. In general, both errors smoothly approach zero by the desired cutoff time of 2.22 hrs. However, over the course of the simulations, the position errors extend far beyond the initial errors. In particular, the in-plane subsystem has position errors as large as 9000 m. Additionally, the final position errors show residual errors of roughly 1.7 m for the in-plane subsystem, while the cross-track subsystem has tighter convergence with 0.3 m of final error. Since the dynamics are unperturbed, this final error appears to result from the discrete control update step.

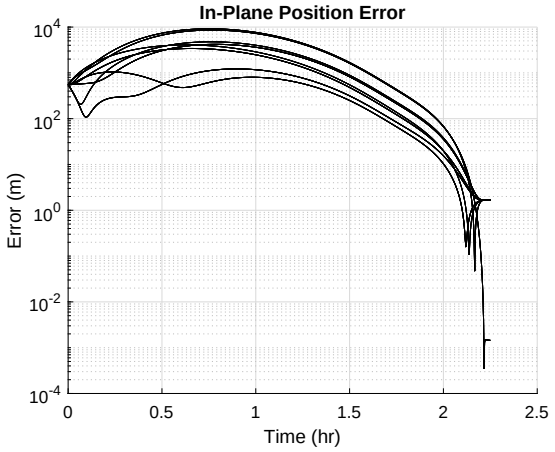

Fig. 6.7: LVLH in-plane state errors when the controller is applied to the unperturbed LVLH dynamics

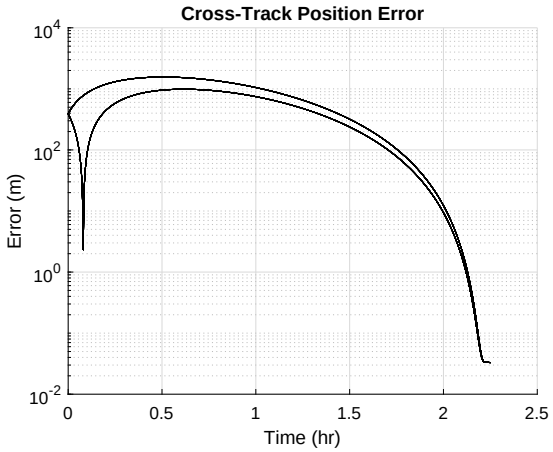

Fig. 6.8: LVLH cross-track state errors when the controller is applied to the unperturbed LVLH dynamics

The control magnitudes are shown in Figures 6.9 and 6.10 for the in-plane and cross-track controls, respectively. For the in-plane control, the control magnitudes oscillate over the course of the simulations but not to the extent that it can be considered chatter. As for the cross-track control, there is a single major change in the control magnitude and the inputs are generally smooth over the full simulation.

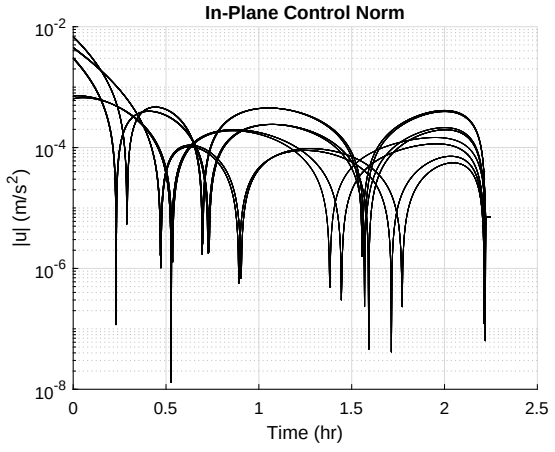

Fig. 6.9: LVLH in-plane controls when the controller is applied to the unperturbed LVLH dynamics

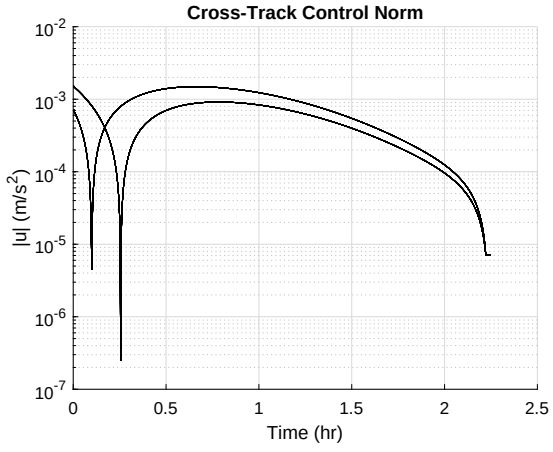

Fig. 6.10: LVLH cross-track controls when the controller is applied to the unperturbed LVLH dynamics

6.4.2 Perturbed CW Dynamics

The second scenario was performed with matched perturbations added along the Y-axis. These perturbations represent the effects of drag due to two spacecraft with different ballistic coefficients. In practice, the drag perturbations affect each LVLH axis but the impact is primarily felt along the Y-axis.

The ballistic coefficient C_b, defined for this work as $\frac{C_D A}{m}$ although it can also be defined as the inverse of this ratio, of a spacecraft defines the effect of drag on a spacecraft with a larger C_b meaning that the spacecraft experiences more drag effects. For this scenario, an exaggerated difference was used to estimate the drag effects with the ballistic coefficient of the reference spacecraft $C_{b,A}$ being 25 times the ballistic coefficient of the spacecraft being controlled $C_{b,B}$.

Given the positions r_A and r_B with ballistic coefficients $C_{b,A} = 25C_{b,B}$, the accelerations due to the dynamics and drag are found as

$$\dot{r} = \bar{f}(r, C_b) \tag{6.45}$$

with the difference in acceleration in the ECI frame being

$$\delta\dot{r} = \dot{r}_A - \dot{r}_B \tag{6.46}$$

With the transformation matrix $T_{LVLH}(r)$ which converts the ECI to LVLH frame based on the state r, the acceleration difference in the LVLH frame can be found as

$$\delta\dot{r}_{LVLH} = T_{LVLH}(r_A)\delta\dot{r} \tag{6.47}$$

Examining the effect of the difference in LVLH acceleration along a spacecraft's orbit in LEO, the Y-axis difference is approximately $5 \times 10^{-6}\,\mathrm{m/s^2}$ with the difference along the X- and Z-axes being two orders of magnitude smaller. Based on this, the CW dynamics are perturbed by $5 \times 10^{-6}\,\mathrm{m/s^2}$ along the Y-axis with no perturbations in the X- and Z-axes with

a disturbance rejection term $D = 7 \times 10^{-6}\,\mathrm{m/s^2}$ used with both the Y- and Z-axes controls. This satisfies the requirement that the perturbations for the predefined-time controller are matched. A total of 50 simulations are run and the tuning functions are optimized using three additional basis functions.

In Figures 6.11 and 6.12, the state errors for the in-plane and cross-track subsystems, respectively, are shown. Overall, trends for both look similar to the unperturbed results, although the maximum error in both is higher than was seen with the unperturbed dynamics. The cross-track subsystem was not perturbed for these simulations, so the change in error is a result of the tuning function optimization. The in-plane subsystem is perturbed and the control law was optimized so both likely contributed to the higher error.

Examining the errors at the end of the simulations, the in-plane position errors range from 4.2 m to 9.0 m, which is notably higher than the final errors of the unperturbed simulations. This indicates that the controller is unable to fully reject the perturbations, likely due to the sampling of the controller. The final errors for the cross-track error are not notably different from the unperturbed dynamics, with the exception of a single outlier that ends with an error on the order of 10^{-4} m.

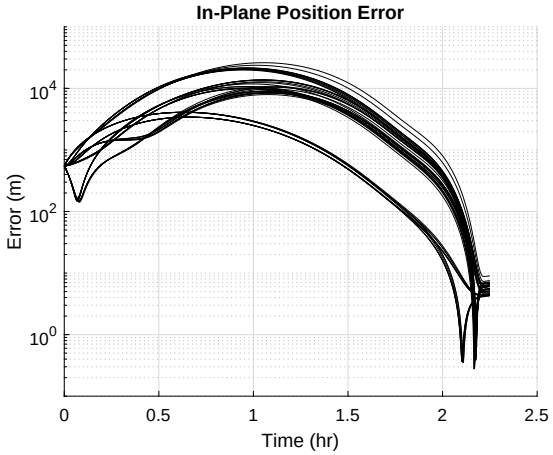

Fig. 6.11: LVLH in-plane states error with a constant, matched disturbance acting along the Y-axis

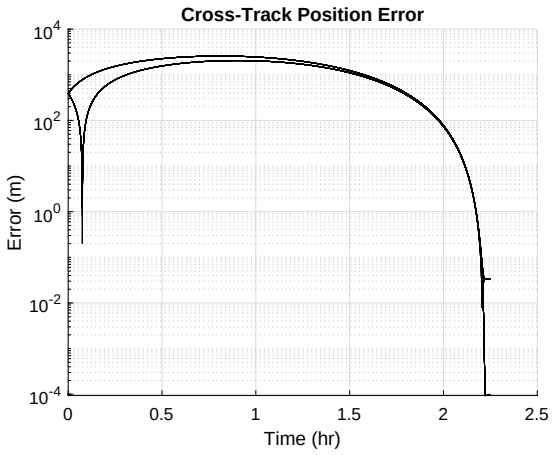

Fig. 6.12: LVLH cross-track states error with a constant, matched disturbance acting along the Y-axis

Figures 6.13 and 6.14 show the related control plots for the in-plane and cross-track subsystems, respectively. The first item of note is that the in-plane control has significantly more variation than was seen with the unperturbed dynamics as a result of the tuning function's optimization. Additionally, the process shows the desired benefits of the optimization. The initial control magnitude for the unperturbed dynamics is found to be as high as $6.8 \times 10^{-3}\,\mathrm{m/s^2}$ while the optimized approach has a lower maximum initial control of $3.7 \times 10^{-3}\,\mathrm{m/s^2}$. However, the optimization has the effect of "flattening" the control profile, as the control magnitudes are generally higher over the remainder of the simulations. For the unperturbed dynamics, the controls max out between $4 \times 10^{-4}\,\mathrm{m/s^2}$ and $5 \times 10^{-4}\,\mathrm{m/s^2}$, ignoring the initial controls. With the perturbed dynamics, however, there are many simulations with control magnitudes higher than this. Thus, the optimization is able to reduce the maximum instantaneous control by raising the control at other points along the trajectory.

The cross-track control plot (Figure 6.14) shows a similar effect with the initial control magnitudes being reduced via the optimization. However, the resultant rise in the remainder of the control trajectory eliminates any gains. As discussed before, this is likely a result of the Brunovsky transformation where the optimization of the tuning function is unable to account for the additional control.

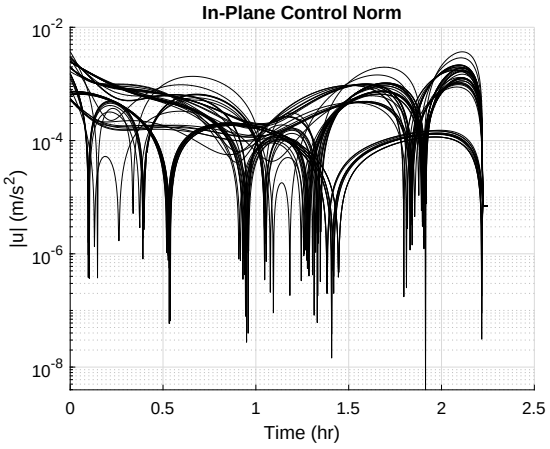

Fig. 6.13: LVLH in-plane control with a constant, matched disturbance acting along the Y-axis

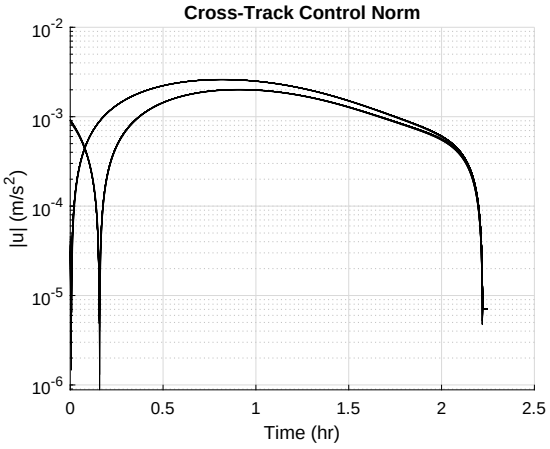

Fig. 6.14: LVLH cross-track control with a constant, matched disturbance acting along the Y-axis

6.4.3 Two-Body Dynamics

A final set of simulations was run using the full 2-body dynamics with J2 and drag perturbations and the two input finite-time backstepping control for the in-plane dynamics. A perturbation bound of $D = 1 \times 10^{-4}\,\mathrm{m/s^2}$ was used with a convergence time of 2.2 hrs. Since the control law was seen to struggle with fully converging to the origin with the single-input in-plane control, the dual-input in-plane control was used with the two-body dynamics. This has the additional effect of fully satisfying the matched disturbance requirement of the control law. A total of 50 simulations were run and the tuning functions were optimized with three additional basis functions.

The error plots for the in-plane and cross-track states are shown in Figures 6.15 and 6.16, respectively. The in-plane plots look similar to the cross-track results, now that each axis is being controlled as if it were an independent subsystem comprised of a double integrator. Additionally, both the maximum and final errors are lower with the dual-input in-plane control. The maximum errors never exceed 4000 m while the final errors are all between 1 m and 4 m. While lower than the perturbed CW results, this is still significantly larger than the ideal exact convergence to the origin that the controller can theoretically achieve. Now that the cross-track dynamics are perturbed as well, the position errors show a greater variance. This is most prominent in the final errors, which lie between 1.6 m and 0.3 m.

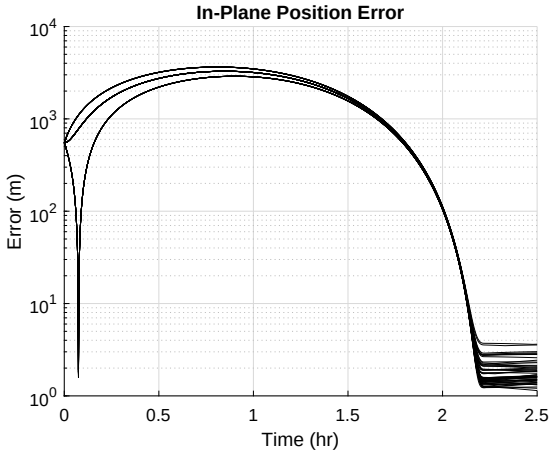

Fig. 6.15: In-plane position errors with perturbed two-body dynamics and dual-input control.

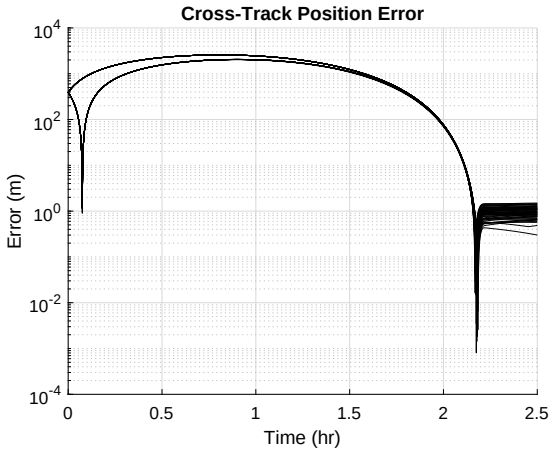

Fig. 6.16: Cross-track positon errors with perturbed two-body dynamics.

The control plots for the in-plane and cross-track subsystems are shown in Figures 6.17 and 6.18, respectively. Again, the in-plane controls show a different control profile now that the subsystem is being controlled as two independent double integrators. Careful inspection shows that there are now oscillations in the control profiles for both the in-plane and cross-track subsystems which appear as wider bands around the control trajectories. However, these are relatively small changes in magnitude and generally oscillate slower than the control update period. This addresses the main concerns with chatter where the controller calls for the actuators to make large, rapid changes.

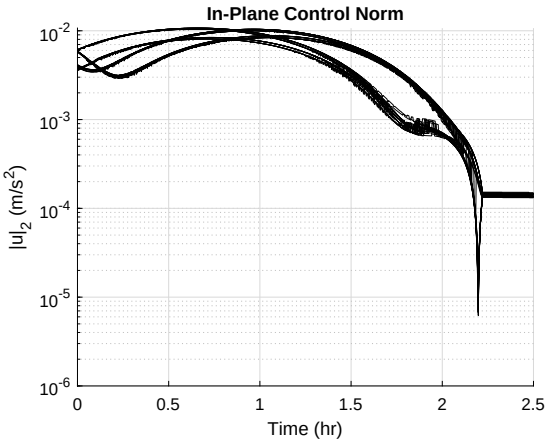

Fig. 6.17: In-plane control for the perturbed two-body dynamics and dual-input control.

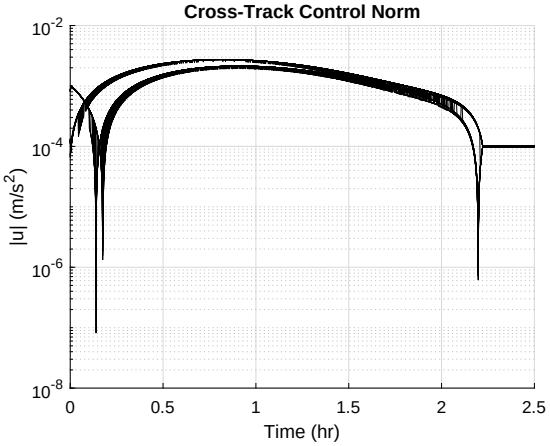

Fig. 6.18: Cross-track control for the perturbed two-body dynamics.

6.5 Conclusion

Predefined-time backstepping [3] is a method which allows for the formulaic construction of a predefined-time controller. The controller assumes dynamics in a specific form similar to an integrator chain but allowing nonlinearities in each state that do not involve any higher states. Additionally, it assumes a single-input control which is applied to the highest state and that perturbations only affect the system through this state as well.

The use of the Brunovsky form allowed for the CW dynamics to be used with the controller, despite the seeming incompatibility, by dividing the dynamics into separate groups which could be controlled individually. A method to minimize the maximum instantaneous control method was introduced and shown have an effect but to not work reliably with the system converted into the Brunovsky form due to the inclusion of the transformation terms.

The system and controller were tested using CW dynamics perturbed along the Y-axis and the full two-body dynamics with J2 and drag perturbations. Overall, the predefined-time controller was able to successfully control a spacecraft to a desired trajectory when the matched disturbance assumption is met. However, the lack of bounds on either the states or control are a limiting factor.

CHAPTER 7

FINITE-TIME CONTROL VIA CONTROL SWITCHING FUNCTIONS

This chapter examines three control laws whose behavior is determined by a set of switching conditions. The first control law is inspired by minimum-time control of a double integrator system. The second and third control laws are variants of sliding mode control. A general background of minimum-time and sliding mode control are presented in Section 7.1 followed by the description and analysis of each of the three control laws in Sections 7.2, 7.3, and 7.4.

7.1 Background

A switching surface naturally arises in the minimum time control of the double integrator. The optimal control consists of the maximum control magnitude with alternating directions [10]. Furthermore, the states where the control directions switch can be plotted to determine areas divided by a switching surface. The control inputs are then determined by the current area the system is in and change as the system passes through the surface. Since the control law results in the system alternating between the extremes of the allowable control inputs, it falls into a category of controllers known as "bang-bang" controls [81, 82].

For the double integrator system

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned} \tag{7.1}$$

with allowable controls $-1 \leq u \leq 1$, the switching surface is found to be $x_1 = -\frac{1}{2}x_2|x_2|$ [32]. If the system is above this surface then a control of $u = -1$ is applied while a control of $u = +1$ is applied if the system is below the surface. When the system reaches the surface, the sign of the control changes which drives the system to the origin.

Another control methodology is sliding mode control [38] which defines a surface that naturally drives the system toward the origin. The control law is then developed to drive the system to the surface in finite-time and then along the surface to the origin. In general, sliding mode control guarantees asymptotic convergence along the surface to the origin but some approaches, including the ones used in Sections 7.3 and 7.4, guarantee finite-time convergence. The controller is robust to disturbances but can suffer from chatter that occurs in the presence of perturbations since the system cannot exactly stay on the designated surface. This is shown in the following example.

Example 7.1.1 ([83])**.** Consider the system

$$\begin{aligned} \dot{x} &= x^3 + z \\ \dot{z} &= u + \delta(t) \end{aligned} \tag{7.2}$$

where $\delta(t) \in \mathbb{R}$ is some unknown perturbation with a known bound $|\delta(t)| \leq \Delta$.

If the behavior of the x state was such that

$$\dot{x} = -x \tag{7.3}$$

then the origin of X-subsystem would have exponential stability. Substituting the system dynamics into the desired dynamics gives

$$0 = x^3 - x + z \tag{7.4}$$

Now, if we define σ such that

$$\sigma = x^3 - x + z \tag{7.5}$$

then when $\sigma = 0$, the origin of the X-subsystem is exponentially stable.

Introducing a Lyapunov candidate function

$$V = \frac{1}{2}\sigma^2 \tag{7.6}$$

with

$$\begin{aligned}\dot{V} &= \sigma\dot{\sigma} \\ &= \sigma(3x^5 + 3x^2z + u + \delta(t) + x^3 + z)\end{aligned} \tag{7.7}$$

Selecting

$$u = v - 3x^5 - 3x^2z - x^3 - z \tag{7.8}$$

gives

$$\dot{V} = \sigma(v + \delta(t)) \tag{7.9}$$

While $\delta(t)$ is unknown, we can use the bound $|\delta(t)| \leq \Delta$ to ensure that $\dot{V}$ is negative definite by selecting $v = -k\ \text{sign}(\sigma)$ with $k > \Delta$. Then

$$\begin{aligned}\dot{V} &= \sigma(-k\ \text{sign}(\sigma) + \delta(t)) \\ &= -k|\sigma| + \delta(t)\sigma \\ &\leq -k|\sigma| + \delta(t)|\sigma| \\ &= (\delta(t) - k)|\sigma| \\ &\leq -\alpha|\sigma| \\ &= -\alpha(\sigma^2)^{\frac{1}{2}} \\ &= -\sqrt{2}\alpha V^{\frac{1}{2}}\end{aligned} \tag{7.10}$$

where $\alpha = -\Delta + k > 0$, due to the previous choice of k. Since $\sqrt{2}\alpha > 0$ and $0 < \frac{1}{2} < 1$, V reaches 0 in finite time.

Thus, the system reaches the σ surface in finite time and the origin is exponential stability as the system moves along the surface.

$\triangle$

The inclusion of the $k\ \text{sign}(\sigma)$ term in the control law results in a discontinuous control as the perturbations force the system off the σ surface. In particular, this can result in chattering behavior where the $k\ \text{sign}(\sigma)$ rapidly jumps between positive and negative k. This can result in large amounts of control usage since the control applied by the $k\ \text{sign}(\sigma)$ term applies

the same control magnitude regardless of the magnitude of the σ error. Additionally, this control law assumes an actuator that can respond instantaneously, which is not reasonable for physical systems.

One approach to minimize these limitations is to use a continuous function to approximate the sign function. Some common selections are the sigmoid function [83]

$$\mathrm{sig}_\gamma(\sigma) = \frac{1 - e^{-\sigma\gamma}}{1 + e^{-\sigma/\gamma}} \tag{7.11}$$

the continuous saturation function

$$\mathrm{sat}_\gamma(\sigma) = \begin{cases} 1 & \sigma/\gamma > 1 \\ \sigma/\gamma & -1 \leq \sigma/\gamma \leq 1 \\ -1 & \sigma/\gamma < -1 \end{cases} \tag{7.12}$$

and the hyperbolic tangent

$$\tanh_\gamma(\sigma) = \tanh \gamma\sigma \tag{7.13}$$

Each of these can be scaled such that as $\gamma \to \infty$, the function better approximates the sign function. While these approximations are not guaranteed to satisfy the conditions for finite-time stability, they generally are able to maintain the system within a neighborhood of the origin, with the neighborhood decreasing as the approximation approaches $\mathrm{sign}(\sigma)$ [83].

Techniques such as uniform ultimate boundedness [38, 84, 85], which guarantees finite-time convergence to a neighborhood of the origin, can be used to compare performance of the nominal control law with the switching function approximation. Alternatively, it may be possible to redesign the control law to naturally attenuate chatter without losing the finite-time guarantee, as seen in [86].

7.2 Switching Surface Based Control

For minimum-time control of the double integrator, it is known that a surface can be generated such that the system will converge to the origin along that surface when acted upon by a certain constant control. This section modifies that approach by allowing the switching surfaces and associated control magnitudes to be optimized to minimize the maximum instantaneous control used. This is applied to the rendezvous problem using the Brunovsky transformed CW dynamics.

Assuming control inputs are allowed along each axis of the system, the CW dynamics can be separated into three subsystems along the X-, Y-, and Z-axes. In the Brunovsky form, the general form of the dynamics of each subsystem are expressed as

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= v \end{aligned} \tag{7.14}$$

Assuming a constant control v, this allows for the simple analytic solution

$$\begin{aligned} z_1(t) &= z_1(t_1) + z_2(t_1)t + \frac{1}{2}vt^2 \\ z_2(t) &= z_2(t_1) + vt \end{aligned} \tag{7.15}$$

where $z_1(t_1)$ and $z_2(t_1)$ are the initial states.

We now define the surface such that at time $t_1 = 0$, all states $z_1(t_1)$ and $z_2(t_1)$ which lie on the surface will be driven to the origin by a constant control v. This can be found by assuming $z_1(t) = z_2(t) = 0$ and eliminating t to get a surface in terms of $z_1(t_1)$, $z_2(t_1)$, and v which gives

$$z_2^2(t_1) = 2z_1(t_1)v \tag{7.16}$$

Examining the form of this solution, we see that $\text{sign}\,(z_1(t_1)) = \text{sign}(v)$ must be true for a meaningful surface. Furthermore, from the analytic solution of $z_2(t)$, we know that if we

assume the desired final condition $z_2(t) = 0$, then

$$z_2(t_1) = -vt \tag{7.17}$$

Assuming that we are only interested in positive time solutions, then we also have the condition $\text{sign}\left(z_2(t_1)\right) = -\text{sign}(v)$. This provides a sufficient number of constraints on $\text{sign}(v)$ that we can express the surface in terms of the magnitude of v without needed to explicitly consider the direction. This gives the final surface as

$$S(v) = \left\{(z_1, z_2) \in \mathbb{R} \times \mathbb{R} | z_2 = -\text{sign}(z_1)\sqrt{2|z_1 v|}\right\} \tag{7.18}$$

Note that this surface divides the z_1-z_2 plane into two distinct portions such that any point which is not on the surface itself is either above or below the surface relative to the z_2 axis. The control applied to the system can then be determined by the system's state relative to the surface, as shown in the following section.

7.2.1 Open-Loop Control

Assume that the system has an initial condition (z_1, z_2) and a control law which first applies a control of magnitude $\nu_1 \in \mathbb{R}_{>0}$ and then switches to apply a control of magnitude $\nu_2 \in \mathbb{R}_{>0}$. The control directions and switching condition are defined as follows.

If the point (z_1, z_2) is on the surface $S(\nu_2)$, then an applied control of magnitude ν_2, in the proper direction, will drive the system along the surface towards the origin. Accounting for the direction, the necessary control v_2 is then found as

$$v_2 = -\text{sign}(z_2)\nu_2 \tag{7.19}$$

which is applied until the system reaches the origin. If we define the initial time for this condition to be t_s, then the time that the system converges to the origin is found to be

$$t_f = t_s + \frac{-z_2}{v_2} \tag{7.20}$$

If the initial condition is not on $S(\nu_2)$, then a control of v_1 with magnitude ν_1 is first applied to the system until the surface is reached. The switching surface function $s : \mathbb{R} \times \mathbb{R}_{>0} \to \mathbb{R}$ which determines the x_2 state of the surface given the x_1 point and control magnitude ν is calculated as

$$s(x_1, \nu) = -\operatorname{sign}(x_1)\sqrt{2|x_1\nu|}. \tag{7.21}$$

Then v_1 is then found as

$$v_1 = \begin{cases} \nu_1 & z_2 < s(z_1, \nu_2) \\ -\nu_1 & z_2 > s(z_1, \nu_2) \\ 0 & z_2 = s(z_1, \nu_2) \end{cases} \tag{7.22}$$

where the first condition is applied when the system is below the surface, the second condition is applied when the system is above the surface, and the third condition represents when the system is already on the surface.

With v_1 determined, the time until the switching surface is reached can be calculated, assuming a switching control magnitude ν_2 has been chosen. Given the initial conditions of (z_1, z_2) at time t_0, Equations 7.15 and 7.16 can be combined and solved for the switching time t_s as a function of v_2 which gives

$$t_s = t_0 + \frac{-2z_2(v_1 - v_2) \pm \sqrt{(4v_2z_2^2 - 8v_1v_2z_1)(v_2 - v_1)}}{2(v_1^2 - v_1v_2)} \tag{7.23}$$

While the direction of v_2 is not known, the magnitude is known to be equal to ν_2. With two possible directions for v_2, this results in four possible solutions for t_s, two for the positive direction and two for the negative direction. The values of $v_2 = \pm\nu_2$ and corresponding times t_s are inserted into Equations 7.15 and 7.16 with the solution being the pair that satisfies all the conditions. Applying the control v_2 to the system from t_0 to t_s will then drive it to the surface.

The full open-loop control can now be summarized as follows. Given an initial condition (z_1, z_2) at time t_0, select control magnitudes ν_1 and ν_2 and determine the surface $S(\nu_2)$. If

the system lies off the surface, then the control v_1 can be calculated and applied from time t_0 to t_s to drive the system to the surface with the analytic solution (Equation 7.15) allowing for the state at any $t_0 \leq t \leq t_s$ to be found. At that point, the control v_2 can be determined and applied from time t_s to t_f to drive the system to the origin. The analytic solution can be used to determine any state for $t_s \leq t \leq t_f$ with the initial conditions of the analytic solution corresponding to the system state when it reaches the origin.

7.2.2 Closed-Loop Control

While the open-loop control uses the switching time t_s and final time t_f to determine when to change controls, the closed-loop control continually compares the system's states to the switching surface. As with the open-loop control, an initial state (z_1, z_2) is given and control magnitudes ν_1 and ν_2 are selected. The initial control v_1 is determined by comparing the state to the surface (Equation 7.22) and the system is propagated forward. At each step of the simulation, the current state of each system is checked against the switching surface $S(\nu_2)$. Once a system meets the surface, it switches to the v_2 control with magnitude ν_2 and direction determined by Equation 7.19. Due to perturbations, the system is unlikely to exactly reach the origin but instead will cross the switching surface again. Each time a subsystem state crosses the switching surface, the direction of the v_2 control switches resulting in the closed-loop control law.

7.2.3 Optimization

For this control approach, the variables able to be selected are the two control magnitudes for each independent Brunovsky subsystem. These can then be chosen in order to minimize some desired objective function. For this work, the objective function to be minimized is defined as

$$J = \max(\|u_x(t)\|_\infty, \|u_y(t)\|_\infty, \|u_z(t)\|_\infty) \tag{7.24}$$

where $u_x(t)$, $u_y(t)$, and $u_z(t)$ are the actual controls applied to each axis at time t. This has the effect of minimizing the single highest instantaneous control magnitude seen on any of the three axes.

The optimization is given the initial condition of the system $x_0 \in \mathbb{R}^6$ which is then converted to the Brunovsky state $z_0 \in \mathbb{R}^6$. The control magnitudes for each subsystem are selected within the bounds $0 < \nu_{min} \leq \nu_1, \nu_2 \leq \nu_{max}$ and the open-loop solution for each subsystem is found. From the open-loop solution, the subsystem states and the open-loop controls can be used with the Brunovsky transformation to find the actual controls applied to each axis of the system. The objective value can then be found as the maximum of the ∞-norm of the controls along each axis.

One limitation of this formulation is that there are no tuneable parameters so there is only a single solution, or multiple solutions with identical objective values and no way to select between them. However, if desired, the formulation could be modified to include other desirable outcomes, such as convergence time, with tuning parameters introduced to define the importance of each.

7.2.4 Results

The switching surface control was tested with both the unperturbed CW dynamics as well as the full two-body dynamics with J2 and drag perturbations with dual-input in-plane controls. For each set of dynamics, a total of 50 simulations were selected such that the initial conditions satisfied

$$|x_0| = \begin{bmatrix} 390\,\text{m} & 390\,\text{m} & 390\,\text{m} & 1.5\,\text{m/s} & 1.5\,\text{m/s} & 1.5\,\text{m/s} \end{bmatrix}^T \tag{7.25}$$

The control law was updated at a rate of 20 Hz using a sample and hold approach. The theoretical convergence time was determined from the open-loop control law, and then the closed-loop control was simulated for a period 5% longer than that.

Unperturbed CW Dynamics

In Figures 7.1 and 7.2, the states and controls for a single simulation are shown, respectively. In Figure 7.1, each state is shown with inserts highlighting the chatter that occurs as each state nears the origin. In general, the positions converge smoothly to the origin while the velocity states show sharper changes associated with crossing the switching surface. The abrupt change due to reaching the switching surface is also seen in the control plots in Figure 7.2. Note that the control magnitude for the cross-track dynamics are an order of magnitude smaller than the two in-plane controls. However, the general pattern is the same with control for each state being roughly constant until the switching surface is reached. However, the Brunovsky transformation results in a slight change of the control over time as the states evolve. As the system nears the origin, the control chatters because the discretized time step does not allow the control to switch precisely on the surface.

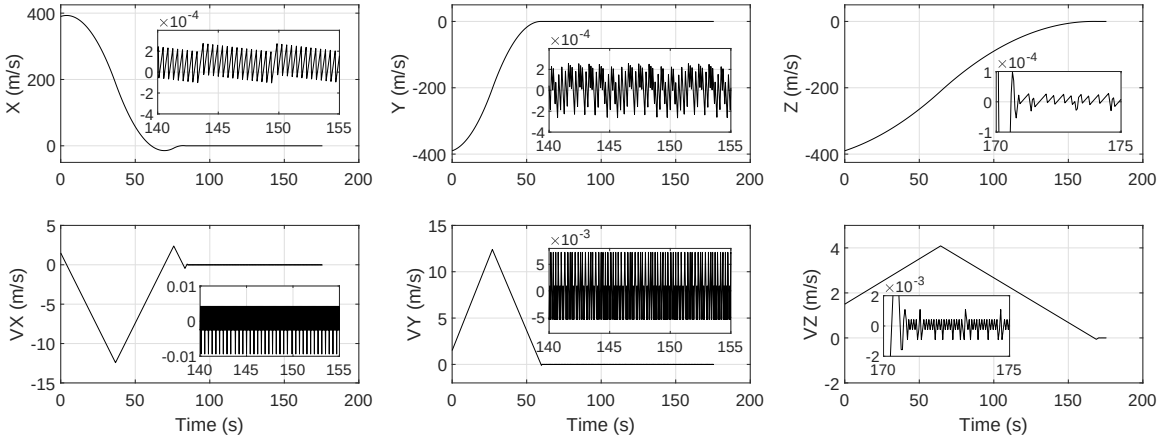

Fig. 7.1: State plots for a single simulation. The inserts show the chattering behavior for each state as the subsystems approach the origin.

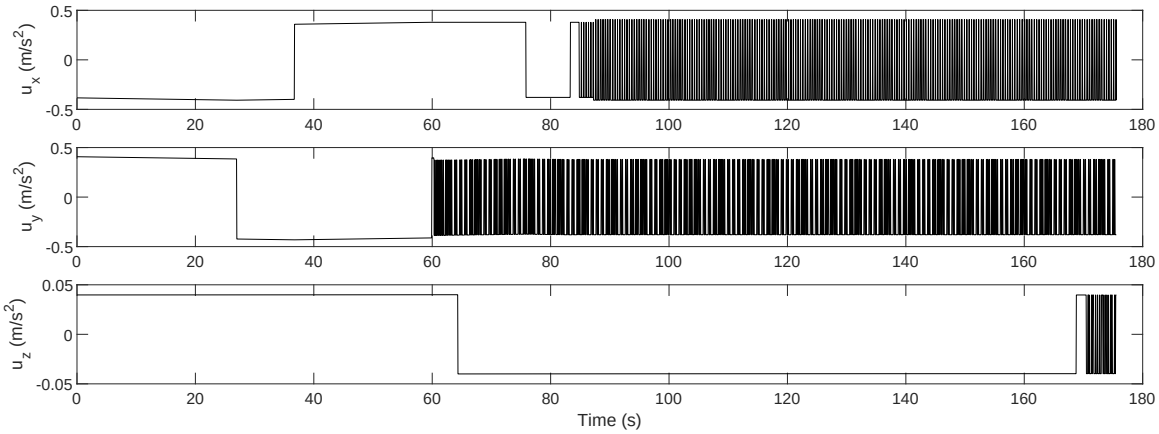

Fig. 7.2: Control plots for a single simulation.

The phase plots for the CW dynamics are shown in Figure 7.3 for each of the subsystems with the initial conditions marked with triangles and the optimized switching surfaces indicated by the dotted lines. Interestingly, while the X- and Y-subsystems are independent in the model dynamics, their responses look nearly identical. This is likely due to the fact that they are coupled through the Brunovsky transformation. The Z-subsystem, which is independent in both the model and full dynamics, shows a much more varied response. However, this indicates the presence of multiple local minima in the objective function.

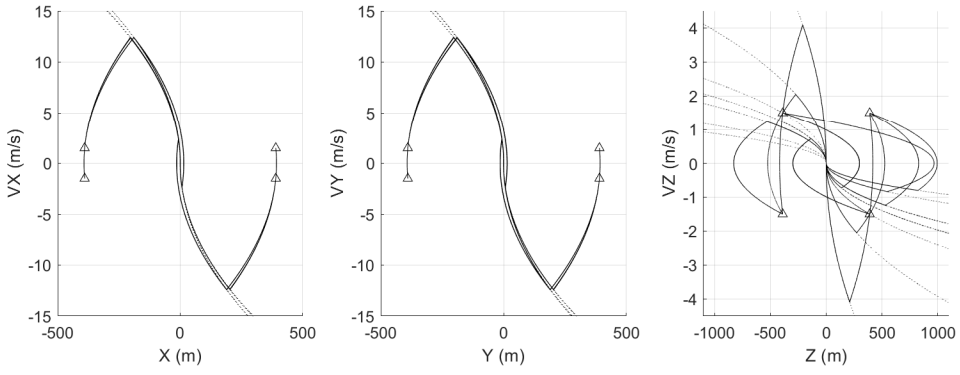

Fig. 7.3: Phase plots for each of the unperturbed CW simulations ran. The initial conditions are indicated by the triangles, while the switching surfaces are shown by the dotted lines.

The position errors for the in-plane and cross-track dynamics are shown in Figures 7.4 and 7.5 respectively. The in-plane dynamics are able to maintain an upper bound on the steady state error of approximately 4×10^{-4} m. However, the the chattering does result in wide ranges of errors. Particularly noticeable in the cross-track errors (Figure 7.5) is the wide range of convergence times. From the plots, it is clear that the convergence time is being driven by the cross-track dynamics. Since the cross-track subsystem has simpler dynamics, the controller can find lower control usage solutions at the cost of longer convergence times.

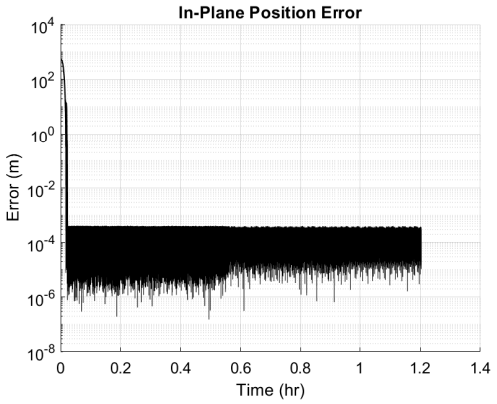

Fig. 7.4: In-plane position error for the unperturbed CW dynamics.

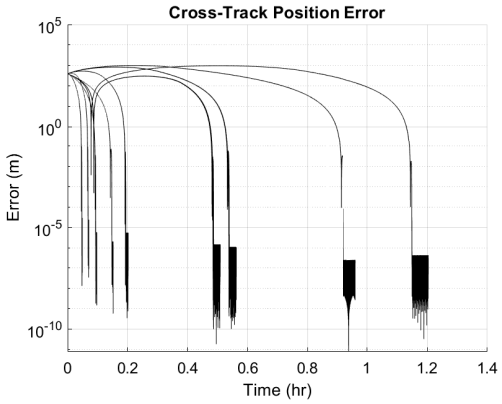

Fig. 7.5: Cross-track position error for the unperturbed CW dynamics.

The corresponding control plots are shown in Figures 7.6 and 7.7 for the in-plane and cross track dynamics, respectively. These plots only show the magnitude of the control in each subsystem, so the changes in direction due to the switching surfaces are not obvious. As a result, the in-plane control shows slight variations initially, but once it begins chattering, the control magnitude is nearly constant. The cross-track control shows larger relative changes in the control magnitude over time as the control from the Brunovsky transformation changes. Additionally, the overall control usage is much smaller in all cases than the corresponding in-plane control magnitude. Note that objective function used with this control law only generates a useful solution due to the use of the Brunovsky transformation. Since the conversion for the Brunovsky control to the actual control is time-varying, the naive approach to simply minimize the Brunovsky control does not necessarily minimize the actual control.

Table 7.1: Reference orbit used for the simulation.

Orbital Element	Value
Semi-major Axis	6 878 000 m
Eccentricity	1×10^{-4}
Inclination	170°
Argument of Perigee	126°
RAAN	0°
True Anomaly	0°

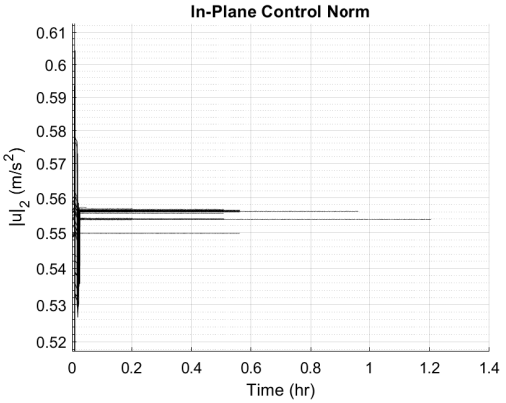

Fig. 7.6: In-plane control for the unperturbed CW dynamics.

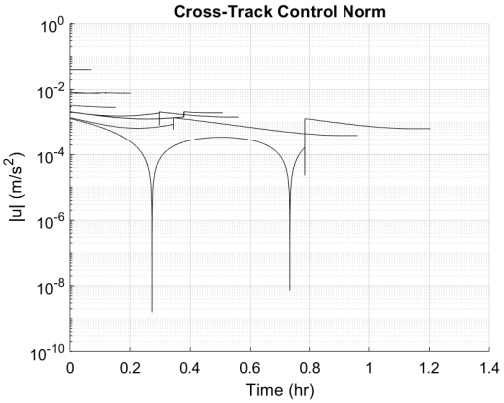

Fig. 7.7: Cross-track control for the unperturbed CW dynamics.

Two-Body Dynamics

The control was also applied to the full two-body simulation with J2 and drag perturbations with orbital parameters as designated in Table 7.1. As with the CW simulations, each simulation ran for a period of time equal to 5% longer than the expected convergence time based on the open-loop solution.

In general, the phase plots (Figure 7.8) look similar to the unperturbed CW results with the X- and Y-axes having similar trajectories, while the Z-axis has multiple solutions from the same initial conditions.

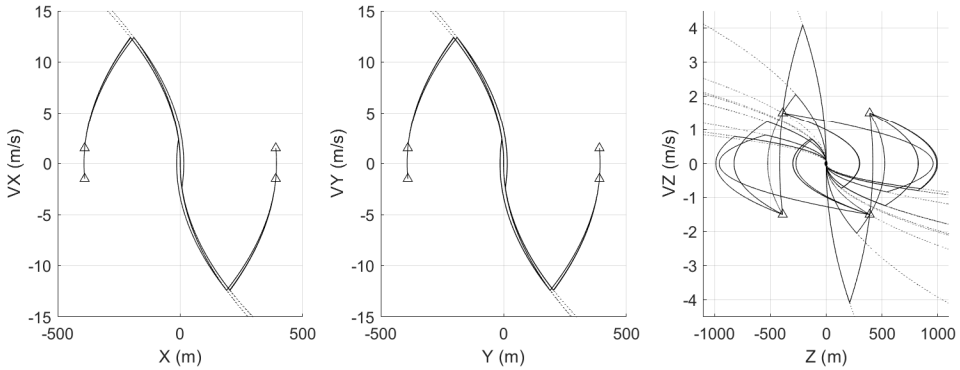

Fig. 7.8: Phase plots for each of the simulations ran using the two-body dynamics. The initial conditions are indicated by the triangles while the switching surfaces are shown by the dotted lines.

The state errors for the in-plane and cross-track subsystems in Figure 7.9 and 7.10 also look broadly similar to the CW results. As before, the in-plane subsystem converges relatively quickly and then chatter about the origin, while the cross-track subsystem takes longer to converge. Additionally, the terminal in-plane error seems to match the CW results, despite the added perturbations. While the terminal cross-track error, generally, has a similar pattern to the CW results, there are a couple simulations which failed to converge to the origin. In total, three simulations had final cross-track errors between 5 m and 8.4 m as the result of the added perturbations. Since the expected perturbations are not accounted for in the control law, the fact that some simulations did not converge is not entirely unexpected. However, in general, the control law is still able to perform satisfactorily in the presence of perturbations.

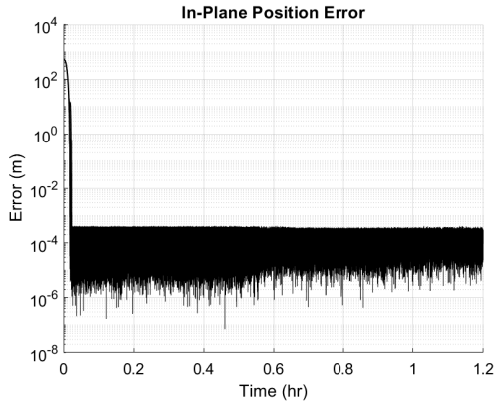

Fig. 7.9: In-plane position error for the two-body dynamics.

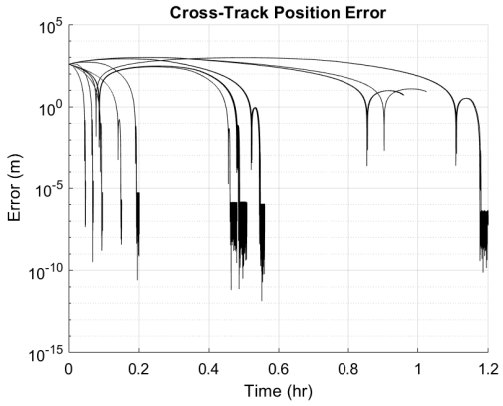

Fig. 7.10: Cross-track position error for the two-body dynamics.

The control plots for the in-plane and cross-track subsystems, Figures 7.11 and 7.12, are also similar to their CW counterparts. Again, the cross-track control magnitudes are significantly smaller than the in-plane control magnitudes, and the relatively rapid convergence of the in-plane dynamics results in the controls chattering with nearly constant magnitude for the majority of the simulation duration.

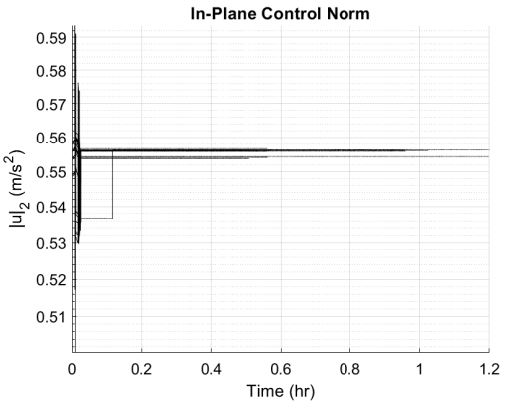

Fig. 7.11: In-plane control for the two-body dynamics.

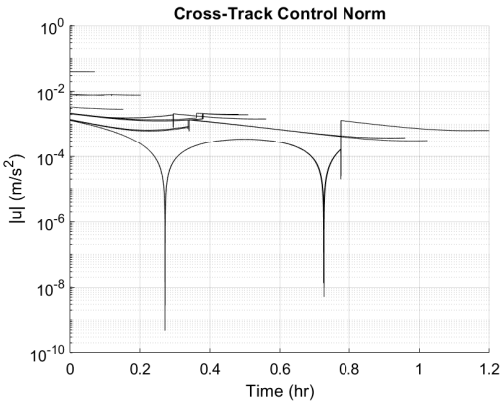

Fig. 7.12: Cross-track control for the two-body dynamics.

7.3 Fractional Power Sliding Mode Control

A finite-time controller, based on sliding mode principles, is presented in [34]. For convenience, we will refer to this controller as a Fractional Power Sliding Mode Control due to the use of the $\lfloor\cdot\rceil^h$ operator, defined as $\lfloor x\rceil^h = |x|^h \operatorname{sign}(x)$.

Assuming the perturbed, double integrator system

$$\begin{aligned} \dot{x}_1 &= x_1 \\ \dot{x}_2 &= u + \delta(t, x) \end{aligned} \tag{7.26}$$

where $\delta(t, x)$ is some bounded, perturbing acceleration, we can introduce the control law as follows.

Theorem 12 ([34, Theorem 2])**.** Given $T_{max} > 0$ with $u \in \mathbb{R}$ and $x \in \mathbb{R}^2$, define a control law as

$$u = -\operatorname{sign}(s)\frac{\alpha_1 + 3\beta_1 x_1^2 + 2k}{2} - \left\lfloor \alpha_2 s + \beta_2 \lfloor s\rceil^3 \right\rceil^{1/2} \tag{7.27}$$

with

$$s = x_2 + \left\lfloor \lfloor x_2\rceil^2 + \alpha_1 x_1 + \beta_1 \lfloor x_1\rceil \right\rceil^{1/2} \tag{7.28}$$

and $\alpha_1/2 = \alpha_2 = \beta_1/2 = \beta_2 = 64/T_{max}^2$ with $|\delta(t, x)| \leq k$ for all $t > 0$ and all $x \in \mathbb{R}^2$. Then, the system 7.26 is globally fixed-time stable with $T(x_0) \leq T_{max}$ for all $x_0 \in \mathbb{R}^2$.

7.3.1 Switching Function

This control law was tested over a range of convergence times T_{max} using both the sign and tanh functions for the switching function. The resulting behavior of the states, control, and actual convergence time were then measured and compared. The simulations were all run using the two-body dynamics with J2 and drag perturbations with the dual-input in-plane controls. As with the previous controller, the controls were applied as a sample and hold with a controller frequency of 20 Hz. As with the switching surface control, the initial

conditions were randomly selected to satisfy

$$|x_0| = \begin{bmatrix} 390\,\text{m} & 390\,\text{m} & 390\,\text{m} & 1.5\,\text{m/s} & 1.5\,\text{m/s} & 1.5\,\text{m/s} \end{bmatrix} \tag{7.29}$$

For the full dynamic simulations, the orbital parameters given in Table 7.1 were used for the reference orbit.

Sign Switching Function

The first set of results use the control law as given in Equation 7.27 over a range of values $30\,\text{s} \leq T_{max} \leq 1000\,\text{s}$. In Figure 7.13 the convergence time is shown as a function of T_{max}. For the purpose of this analysis, the convergence time is defined to be the time at which the position enters and stays within 0.01 m of the origin. With increasing T_{max}, the convergence time also increases although the effect seems to taper off with the high T_{max} values not having proportionally longer convergence times. This can particularly be seen in Figure 7.14 which shows the same convergence times but as a ratio relative to T_{max}. This control law seems to be rather conservative in its selection of T_{max} as the longest ratio is approximately 0.40 of T_{max}. Furthermore, the ratio drops to nearly 0.20 as the T_{max} increases.

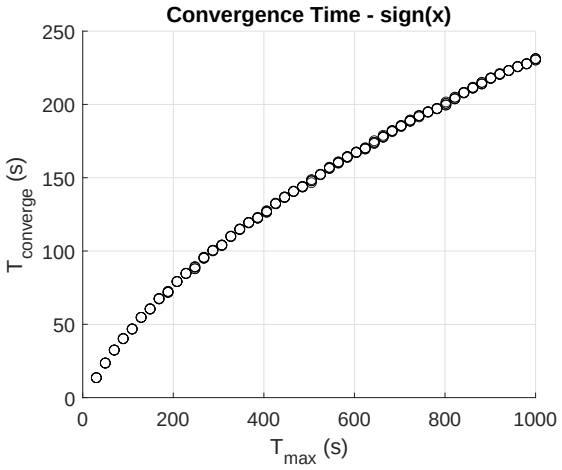

Fig. 7.13: Actual convergence time as a function of T_{max} for the control law given in Equation 7.27.

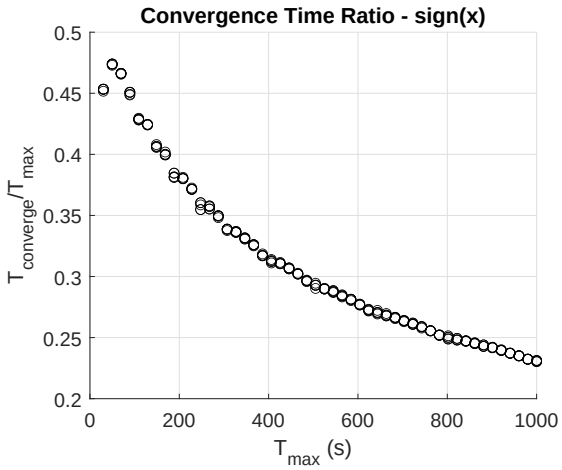

Fig. 7.14: Ratio of actual convergence time to T_{max}.

In Figures 7.15 and 7.16, the log of the state and control magnitude are plotted against the time normalized by T_{max}. For the states, the final position is general between 1×10^{-3} m and 1×10^{-10} m on each axis with the maximum velocity error generally between 1×10^{-2} m/s and 1×10^{-5} m/s. Additionally, there is a clear transition from when the system is approaching the origin and when it is chattering about the origin.

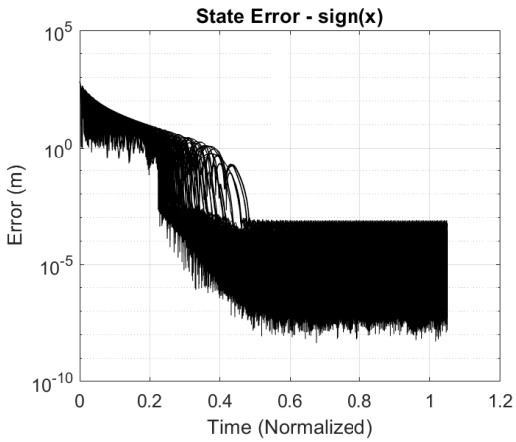

Fig. 7.15: States for all the simulations with the control law from Equation 7.27 on a log scale with the X-axis showing the time normalized by T_{max}.

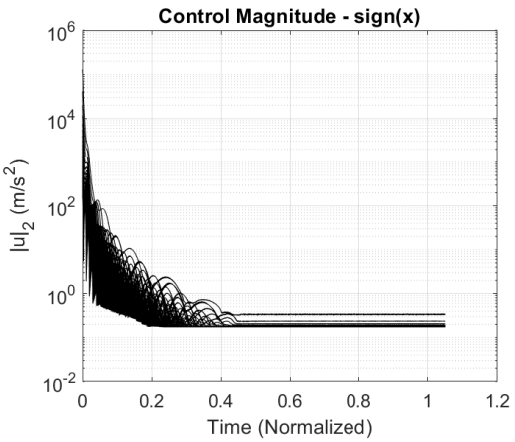

Fig. 7.16: Controls for all the simulations with the control law from Equation 7.27 on a log scale with the X-axis showing the time normalized by T_{max}.

Hyperbolic Tangent Switching Function

For the next portion of the analysis, the hyperbolic tangent function were used in place of the sign function to attempt to minimize the chatter of the control law. This was applied in the form $\tanh(\gamma x)$ with $\gamma \in \mathbb{R}_{>0}$ being a scaling parameter. As γ increased, the slope around the origin steepened and became a better approximation of the sign function.

The impact of γ on the performance of the controller was explored by examining the response of the control law over a range of γ and comparing the final position error and control chatter. A total of 240 simulations were run to describe the behavior of the hyperbolic tangent switching function over a range of 80 values for γ. For each γ value, three simulations were run with T_{max} values of 600 s, 4300 s, and 8000 s and all simulations had an initial condition

of

$$x_0 = \begin{bmatrix} 390\,\text{m} & 390\,\text{m} & 390\,\text{m} & 1.5\,\text{m/s} & 1.5\,\text{m/s} & 1.5\,\text{m/s} \end{bmatrix}^T \tag{7.30}$$

Each simulation was propagated using the unperturbed CW dynamics.

In Figure 7.17, the position error of each simulation at time $t = T_{max}$ is shown. Ideally, this should be 0 for each simulation but due to the hyperbolic tangent approximation and the zero-order hold, some residual error remains. In line with what could be expected, as γ increases, and the hyperbolic tangent better approximates the sign function, the error decreases. The error drops off exponentially with γ meaning that there is a point of diminishing returns where large increases in γ only result in small changes in the final error.

In Figure 7.18, the 2-norm of the control magnitude at time $t = T_{max}$ is shown. Although some variation, due to the difference in the T_{max} parameter, is seen in the plots, some general trends can be observed. For small values of γ, the system never converges to the origin and so there is always some non-zero control being applied to the system by the control law. At the other end, for large values of γ, the behavior of the hyperbolic tangent function exhibits the same chattering behavior that is seen in the sign function. This is reflected in the plot by the jump in control magnitude. Between those two extremes is a range of γ values which result in a control law that is smooth enough to avoid chattering controls but similar enough to the sign function that the control law is still able to control the system.

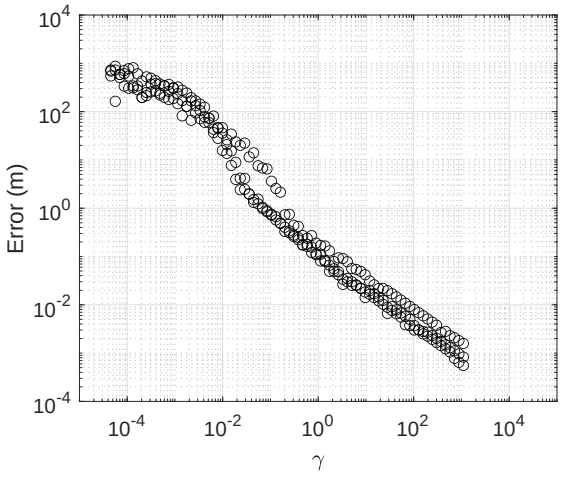

Fig. 7.17: Position error at time $t = T_{max}$ for the control law with a switching function of $tanh(\gamma x)$.

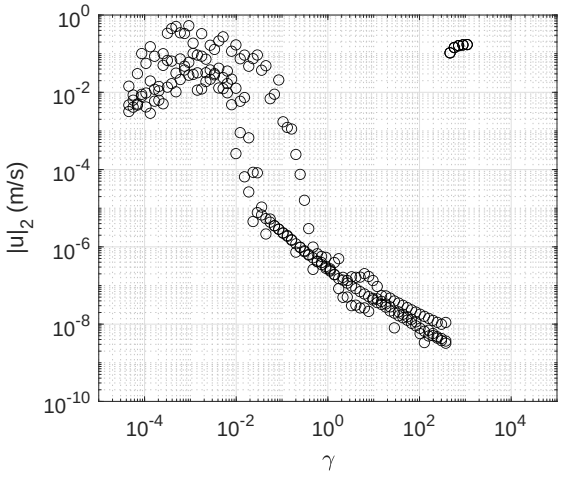

Fig. 7.18: Control magnitude at time $t = T_{max}$ for the control law with a switching function of $tanh(\gamma x)$.

By comparing the two plots, we can select a range of γ that converges by the designated time but does not result in chattering behavior. This appears to approximately be the range $1 \leq \gamma \leq 375$ with the corresponding terminal position errors being below 0.2 m. For this work, a value of $\gamma = 100$ is chosen to balance lower terminal error with non-chattering controls.

In Figures 7.19 and 7.20, the time required for the system to converge to within 0.01 m is shown. Figure 7.19 shows the convergence time in seconds while Figure 7.20 shows the convergence time as the ratio of the convergence time to T_{max}. Similar to the use of the sign function, the convergence time generally increases as T_{max} increases but the rate slows as T_{max} increases. However, the convergence times are notably higher when using the hyperbolic tangent than with the the sign function (Figure 7.13). In Figure 7.20, the ratio of the convergence time to T_{max} shows that the use of the hyperbolic tangent pushes the convergence time closer to, and even over, the T_{max} time. This shows that the inclusion of the hyperbolic tangent nullifies the convergence guarantee of the control law. However, for longer T_{max} times, the control law generally seems to be conservative enough to satisfy the constraint despite the change in switching function.

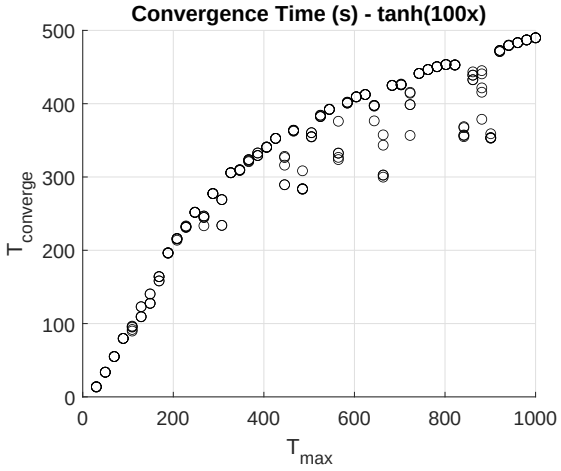

Fig. 7.19: Actual convergence time as a function of T_{max} for the control law using the hyperbolic tangent as the switching function.

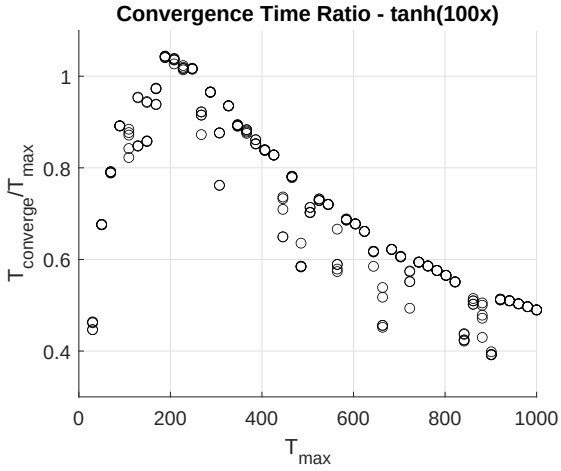

Fig. 7.20: Ratio of actual convergence time to T_{max}.

In Figure 7.21, the states for all the simulations are shown. The maximum final position error for each axis has a rough upper bound of 0.01 m with the best case results being approximately 0.0003 m. This is notably different from the nominal control law where all simulations resulted in less than 0.0001 m of error. For the control profile (Figure 7.22), the control magnitudes at the start of each simulation are roughly identical to the original control law. This results from the fact that the hyperbolic tangent best approximates the sign function in the limit as the magnitude of the argument approaches infinity. For this system, the maximum error, and thus closest approximation, occurs at the beginning of the simulation. As the error decreases, the control law with the hyperbolic tangent is less aggressive since the control law is able to scale with the error. As a result, the final control along all axes has an upper bound of approximately $1 \times 10^{-5}\,\mathrm{m/s^2}$ with many simulations seeing control as small as $1 \times 10^{-7}\,\mathrm{m/s^2}$. This is a significant contrast from the nominal control which never had controls less than 1×10^{-2} due to the chattering behavior.

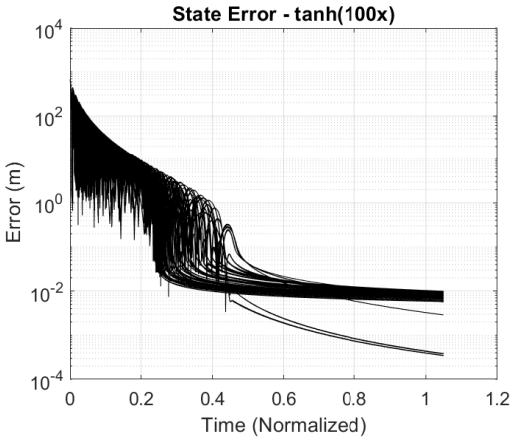

Fig. 7.21: States for all the simulations with the control law from Equation 7.27 on a log scale with the X-axis showing the time normalized by T_{max}.

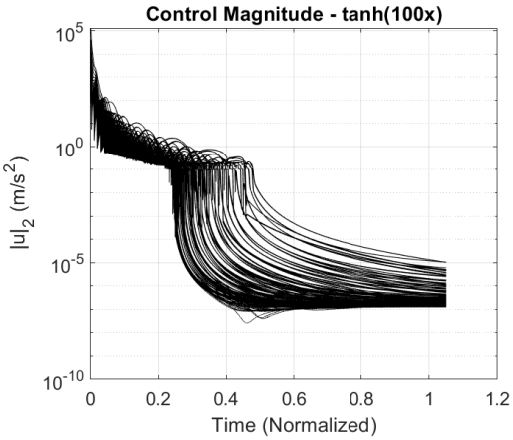

Fig. 7.22: Controls for all the simulations with the control law from Equation 7.27 on a log scale with the X-axis showing the time normalized by T_{max}.

For a final point of comparison, Figure 7.23 shows the control profiles for two simulations, one using the sign function and one using the hyperbolic tangent function. The plots are truncated to focus on the terminal behavior of the control profiles with the plotting beginning at $t = 50\,\mathrm{s}$. Both plots show similar control profiles around the 50 s mark with the start of a sinusoidal or square wave. For the sign function, the control begins chattering with an amplitude of around $0.1\,\mathrm{m/s^2}$ for each axis. However, the oscillations in the control using the hyperbolic tangent dampen out and approach zero.

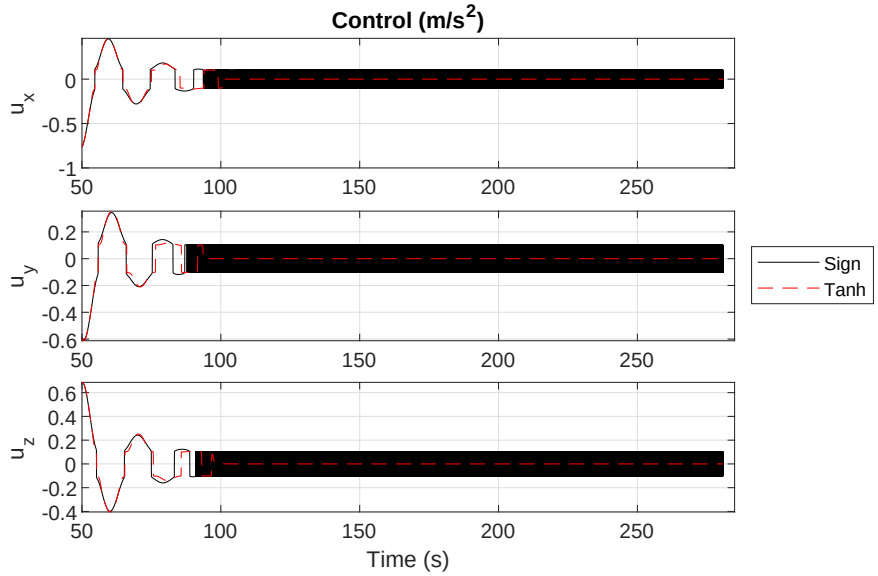

Fig. 7.23: Comparison of the control for two simulations, one using the sign function and one using the hyperbolic tangent function. Note the absence of the chatter in the contol using the hyperbolic tangent.

7.3.2 Results

From the analysis on the switching function, a control law with a convergence time of 400 s and switching function of $\tanh(100x)$ is used to control the system with both the CW and perturbed two-body dynamics. A total of 50 simulations are run, with each randomly initialized to satisfy the initial condition

$$x_0 = \begin{bmatrix} 390\,\text{m} & 390\,\text{m} & 390\,\text{m} & 1.5\,\text{m/s} & 1.5\,\text{m/s} & 1.5\,\text{m/s} \end{bmatrix}^T \tag{7.31}$$

and the orbital parameters are identical to the ones used previously (Table 7.1).

Unperturbed CW Dynamics

The in-plane and cross-track position errors are shown in Figures 7.24 and 7.25, respectively. Interestingly, although the simulations have differing initial conditions, the responses are nearly identical. In both subsystems, the error does not decrease evenly but rather shows a pattern of decreasing oscillations. Once the errors reach about 10^{-2} m, the oscillations die out and the systems reach a roughly steady final error. The convergence time is also shown to be very conservative, with the system converging to the final error well before the 400 s bound.

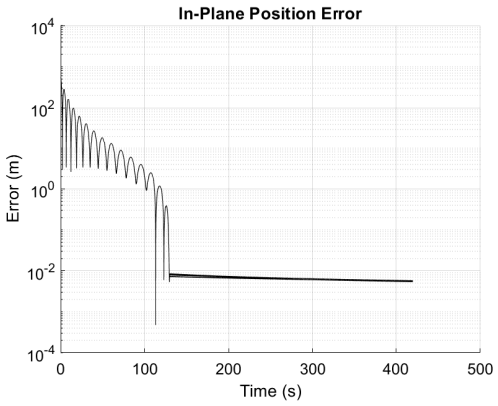

Fig. 7.24: In-plane position error for the unperturbed CW dynamics with the $\tanh(100x)$ switching function and $T_c = 400$.

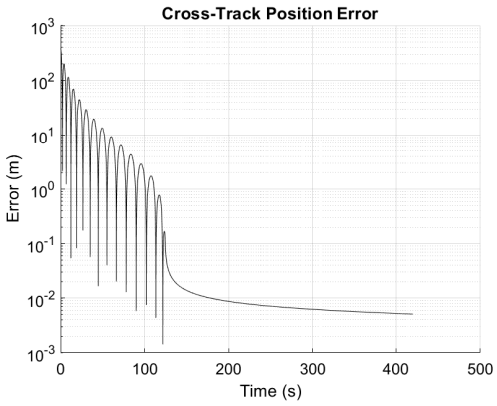

Fig. 7.25: Cross-track position error for the unperturbed CW dynamics with the $\tanh(100x)$ switching function and $T_c = 400$.

The control plots are shown in Figures 7.26 and 7.27 for the in-plane and cross-track controls, respectively. Similar to the error plots, oscillations can be seen in the control before sharply dropping as the system converges to the origin.

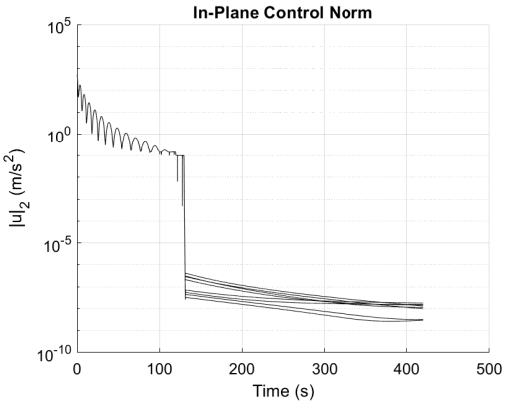

Fig. 7.26: In-plane control for the unperturbed CW dynamics with the $\tanh(100x)$ switching function and $T_c = 400$.

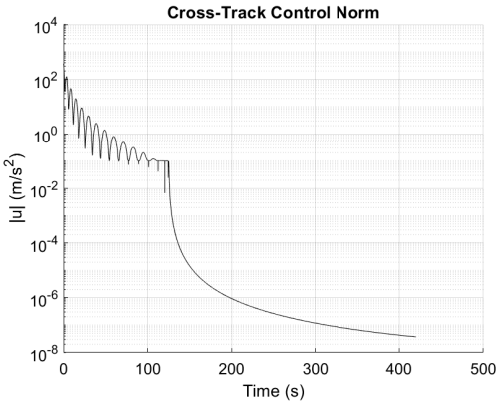

Fig. 7.27: Cross-track control for the unperturbed CW dynamics with the $\tanh(100x)$ switching function and $T_c = 400$.

Two-Body Dynamics

The position errors for the two-body dynamics are shown in Figures 7.28 and 7.29 for the in-plane and cross-track subsystems, respectively. Again, the errors initially oscillate as they decrease, until the oscillations decay away and the system converges to a relatively constant final error. The convergence of the in-plane error is slower with the two-body system than the unperturbed CW dynamics with larger terminal error, which is to be expected. Interestingly, the cross-track error actually appears to converge in about the same time but with a much sharper transition to the terminal error.

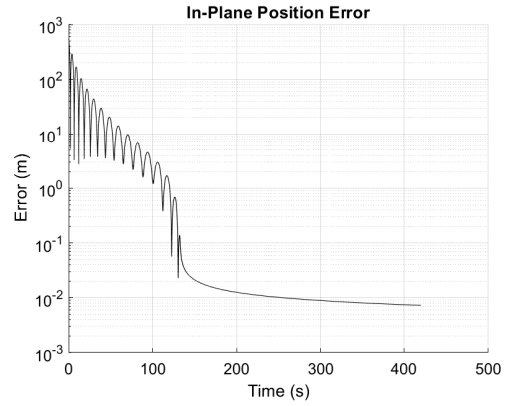

Fig. 7.28: In-plane position error for the two-body dynamics.

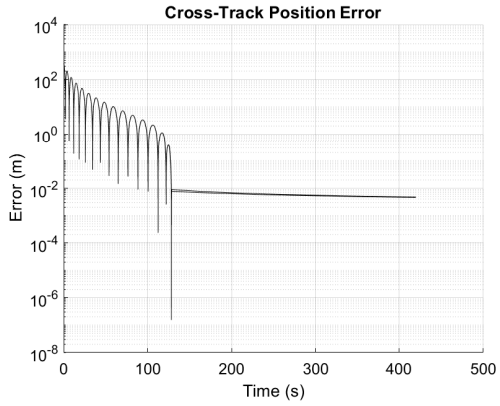

Fig. 7.29: Cross-track position error for the two-body dynamics.

The control plots for the two-body dynamics are shown in Figures 7.30 and 7.31 for the in-plane and cross-track controls, respectively. The basic pattern continues to be the same with decaying oscillations and then a sharper drop-off. However, as a result of the perturbations, the terminal control amounts are larger than what was seen in the unperturbed CW dynamics.

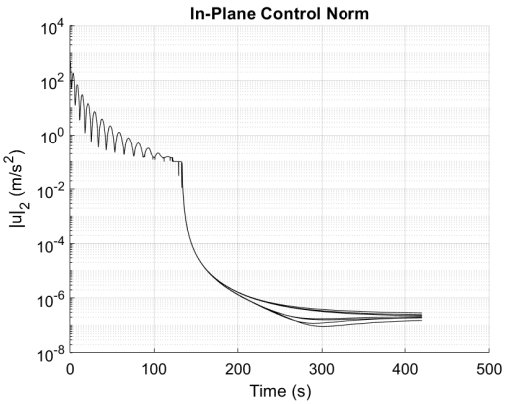

Fig. 7.30: In-plane control for the two-body dynamics.

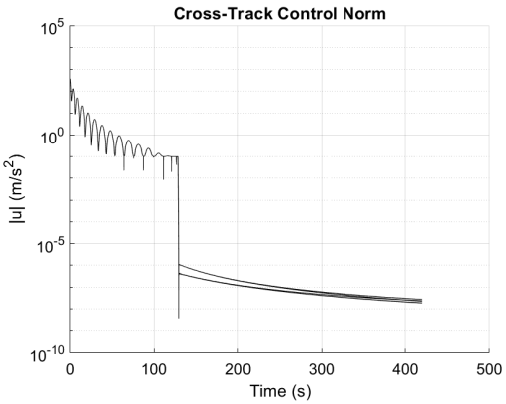

Fig. 7.31: Cross-track control for the two-body dynamics.

7.4 Backstepping Sliding Mode Control

Another approach for a predefined-time controller is developed in [2]. This is based on a backstepping approach but is only applicable for second-order systems which can be transformed into the form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u + \delta(t) \end{aligned} \tag{7.32}$$

with $|\delta(t)| \leq k$ for all $t \geq 0$ and some $k \in \mathbb{R}_{>0}$.

7.4.1 Control Law

The control law requires the introduction of class $\mathcal{K}^1$ and class $\mathcal{W}$ functions, defined as follows.

Definition 7.4.1 ([2, Definition 1])**.** A scalar, continuous function $\kappa : \mathbb{R}_{\geq 0} \to [0, 1)$ is said to be a class $\mathcal{K}^1$ function if

1. it is strictly increasing
2. $\kappa(0) = 0$
3. $\kappa(r) \to 1$ as $r \to \infty$

Definition 7.4.2 ([2, Definition 2])**.** A scalar, continuous function $\omega : \mathbb{R}_{\geq 0} \to [0, 1)$ is a class $\mathcal{W}$ function if

1. $\omega \in \mathcal{K}^1$ is twice differentiable in $\mathbb{R}_{>0}$
2. $\omega'(r) > 0$ for $r > 0$ and $\omega'(0) \in \mathbb{R}_{>0}$
3. $\omega''(r) < 0$ for all $r > 0$

Some examples of class $\mathcal{W}$ given are

1. $\omega(r) = 1 - \exp(-r^q)$
2. $\omega(r) = \frac{2}{\pi}\arctan(r^q)$
3. $\omega(r) = \frac{r^q}{r^q+\alpha}$

with $0 < q < 1$ and $\alpha > 0$.

With a class $\mathcal{W}$ function ω selected, we can now introduce a function $v : \mathbb{R} \to \mathbb{R}$ as

$$v(x) = -\frac{\text{sign}(x)}{\rho_T \omega'(|x|)} \tag{7.33}$$

with $\rho_T > 0$. Additionally, we define two surfaces

$$\begin{aligned} s &= x_2 - v(x_1) \\ s_d &= x_2 + \left\lfloor \lfloor x_2 \rceil^2 - 2 \lfloor v(x_1) \rceil^2 \right\rceil^{1/2} \end{aligned} \tag{7.34}$$

With these preliminaries determined, the control can be defined as [2]

$$u = v(s_d) - k\,\mathrm{sign}(s_d) - s + 2|v(x_1)|v_1'(x_1)\mathrm{sign}(s_d) \tag{7.35}$$

with predefined settling time $T_{max} = 2\rho_T$

7.4.2 Switching Function

When designing this controller, the control parameters ρ_T and k have direct connections to the predefined time and perturbation magnitude, respectively. However, the selection of the class $\mathcal{W}$ function $\omega(r)$ is free but lacks a clear relationship to the system performance. This section explores the selection of $\omega(r)$ and the resultant impact on the CW system. Furthermore, the use of the smooth hyperbolic tangent function in place of the discontinuous sign function is examined.

The CW system was divided into three independent subsystems for each axis using the Brunovsky transformation with an identical controller used for each. The system was propagated using the two-body dynamics with J2 and drag perturbations. The control update rate was set as 20 Hz with the same sample and hold approach used with the other controllers in this chapter. A predefined time of $T_{max} = 300\,\mathrm{s}$ was chosen, resulting in $\rho_T = 150$, and k was selected as $k = 1 \times 10^{-5}$. The class $\mathcal{W}$ function

$$\omega(r) = \frac{r^q}{r^q + \alpha} \tag{7.36}$$

where $0 < q < 1$ and $\alpha > 0$ was selected.

A total of 225 simulations were run with the values of q and α equally spaced such that $q \in [1\times10^{-3}, .999]$ and $\alpha \in [1\times10^{-3}, 1000]$. Each simulation had initial conditions satisfying

$$|x_0| = \begin{bmatrix} 390\,\text{m} & 390\,\text{m} & 390\,\text{m} & 1.5\,\text{m/s} & 1.5\,\text{m/s} & 1.5\,\text{m/s} \end{bmatrix} \tag{7.37}$$

for the relative state in the LVLH frame with the desired final state being the origin. The reference orbit was set as shown in Table 7.1. Similar to the fractional power sliding mode control, each simulation was determined to be converged if the final position error was less than 0.01 m at the end of a 315 s period, otherwise the simulation was termed unconverged.

The results can be seen in Figure 7.32 where the converged simulations are shown with the larger circles as functions of q and α. As a result of the control discretization, the majority of the simulations fail to converge to the origin by the predefined time even though, theoretically, each simulation is guaranteed to converge.

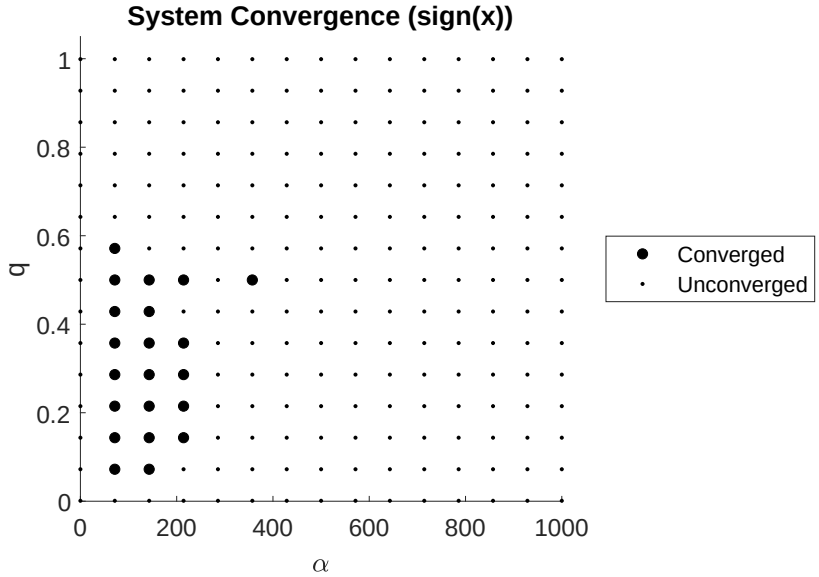

Fig. 7.32: Plot showing the simulations which were stable based on the selection of q and α.

A second set of simulations were run with identical parameters but with the sign(x) functions replaced with the tanh(γx) approximation. An analysis was performed to find a suitable value of γ by running the control law for various values of γ and measuring the

control magnitude and position error at the convergence time T_{max}. Values of $T_{max} = 300\,\text{s}$, $q = .5$, and $\alpha = 200$ were chosen to mirror the parameters presented in Figure 7.32.

In Figure 7.33, the control magnitude is shown. Three distinct sections can be identified. For small values of γ, roughly $\gamma < 1$, the magnitude decreases as γ increases. Then, there is a flat period (approximately $1 \leq \gamma \leq 100$) where the control magnitude is constant. Finally, for values of $\gamma > 100$, chattering behavior occurs and is shown by a marked increase in the amount of control used.

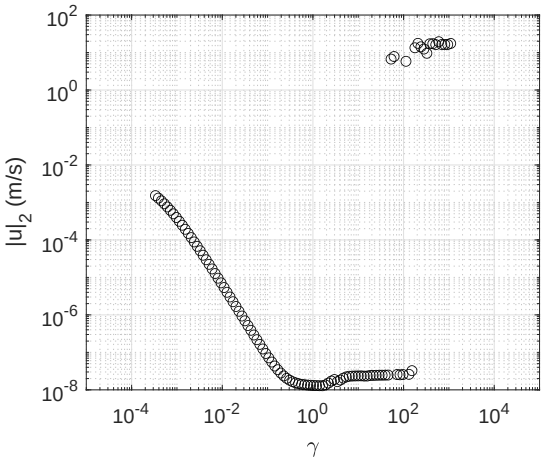

Fig. 7.33: Control magnitude at time $t = T_{max}$ over varying values of γ. Note the large jump around $\gamma = 100$ which indicates the presence of chattering behavior.

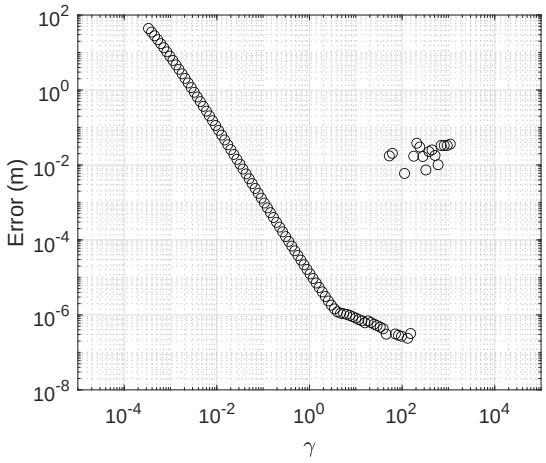

Fig. 7.34: Position error at time $t = T_{max}$ as a function of γ. The terminal error generally decreases until chatter begins which corresponds with an increase in error.

A similar trend can be seen in Figure 7.34 which shows the position error at $t = T_{max}$. There is a steep decline in the error as γ increases to around 2 when the rate of decrease slows. Then around $\gamma = 100$, there is a sharp jump where the position error increases which corresponds with the point where the chattering behavior occurs. This contrasts with the previous control where the the position error continues to decrease as the hyperbolic tangent better approximates the sign function.

Based on these plots, a value of $\gamma = 1$ was chosen for the hyperbolic tangent function. This should result in a non-chattering control with terminal position error on the order of $1 \times 10^{-5}\,\text{m}$. Note that this is a significantly different value than the $\gamma = 100$ which was

chosen for the other approximation. This suggests that there is no general rule in selecting a switching surface approximation but that it must be determined individually for each control law.

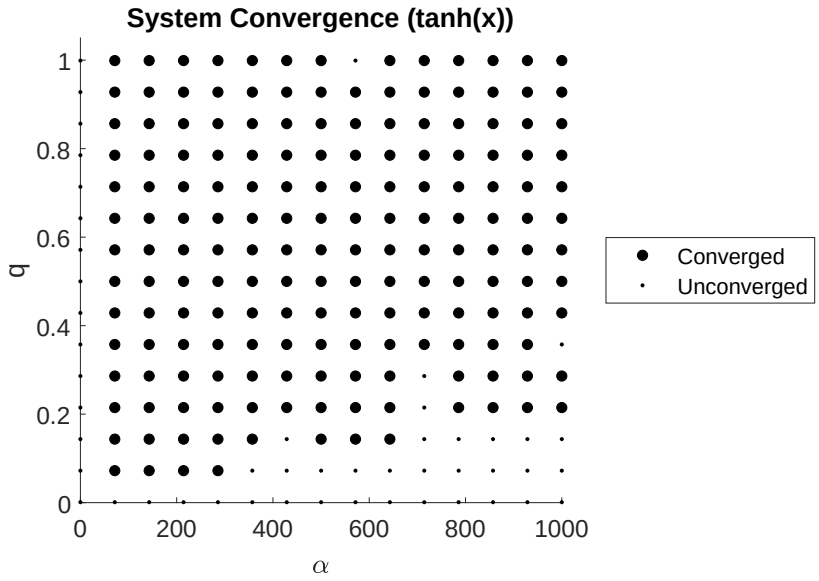

Fig. 7.35: Plot showing the simulations which were stable based on the selection of q and α with $\tanh(x)$ used in place of $\text{sign}(x)$.

The convergence of the system over a range of q and α values and using the hyperbolic tangent are shown in Figure 7.35. Somewhat surprisingly, this change greatly expands the range of parameters for which the system is stable about the origin. In general, the $\tanh(x)$ approximation is applied to switching controls to minimize the presence of chatter in controls with the implicit understanding that there will be an impact in the system stability. This shows that the impact is not, necessarily, a source of instability.

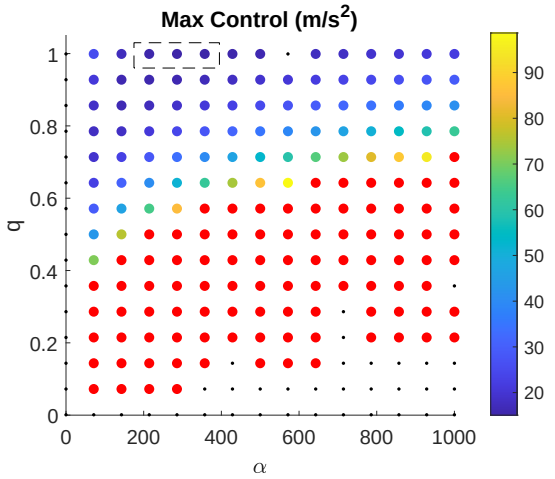

Fig. 7.36: The maximum control as a function of the q and α parameters. For plotting clarity, any maximum control above $100\,\mathrm{m/s^2}$ is indicated by a red dot. The dashed box indicates the area examined in further detail.

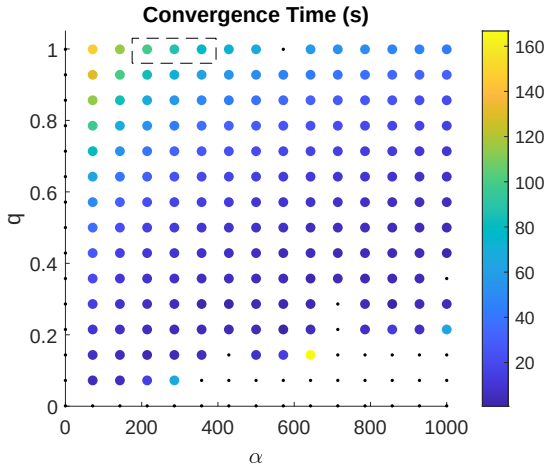

Fig. 7.37: The convergence time as a function of the q and α parameters. The dashed box indicates the area examined in further detail.

7.4.3 Tuning Parameters

In Figures 7.36 and 7.37, the maximum control and convergence time are shown for the varying parameters of q and α. For the maximum control plot, the displayed magnitude is capped at $100\,\mathrm{m/s}$ with any values higher than that displayed in red. However, the maximum control was as high as $5.0454 \times 10^5\,\mathrm{m/s}$ for some simulations. In general, as the value of q increases, the maximum control usage decreases. Similarly, in Figure 7.37, the convergence time is shown. This is defined as the last time the system is more than $0.1\,\mathrm{m}$ away from the origin. In general, the convergence time is very rapid compared to the predefined time of $300\,\mathrm{s}$. Additionally, it appears that the time increases as q increases and as α decreases.

The dashed box in both plots indicates the area that seems to shown the most desirable behavior in terms of minimum control and a convergence time similar to the predefined time. Running another set of 225 simulations over the range indicated by the box gives the results shown in Figures 7.38 and 7.39. Interestingly, there are two unstable simulations within this range of parameters emphasizing the fact that the discretization of the simulation and the use of the $\tanh(x)$ function lacks theoretical stability even though it generally exhibits stable behavior.

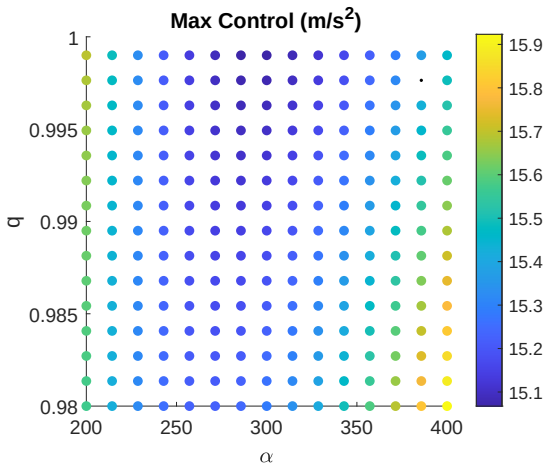

Fig. 7.38: The maximum control as a function of the q and α parameters over a smaller range of values.

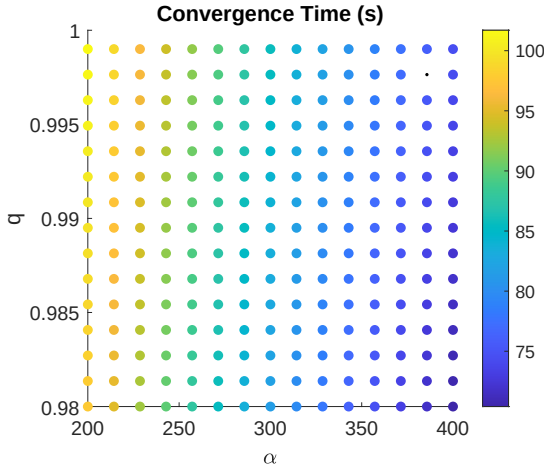

Fig. 7.39: The convergence time as a function of the q and α parameters over a smaller range of values.

In Figure 7.38, the maximum control is shown and an area of lower control can be seen around $q \approx 1$ and $250 \leq \alpha \leq 325$. With the convergence time, the previous pattern holds with the convergence time increasing with decreasing values of α.

We can now examine the terminal error and control of the system compared to the expected values based on the chosen γ. Recall that a value of $\gamma = 1$ was chosen based on the data in Figures 7.33 and 7.34 with expected terminal control of approximately $1 \times 10^{-8}\,\mathrm{m/s^2}$ and terminal error of $1 \times 10^{-5}\,\mathrm{m}$. In Figures 7.40 and 7.41, the control magnitude and position error at time $t = T_{max}$ are shown over the range of q and α values examined. Note that for both plots, the $\log_{10}$ of the magnitude is displayed. For the control (Figure 7.40), most simulations have values between $1 \times 10^{-8}\,\mathrm{m/s^2}$ and $1 \times 10^{-6}\,\mathrm{m/s^2}$. However, as q decreases, the simulations near the unconverged simulations show significantly higher amounts of control. Similarly, the terminal position error (Figure 7.41) shows that for higher values of q the terminal error decreases. While the choice of $\gamma = 1$ produces desirable behavior over a range of q and α values, there is notable interaction between γ, q, and α and it is necessary to consider all three if attempting to satisfy a specific convergence time or control magnitude constraints.

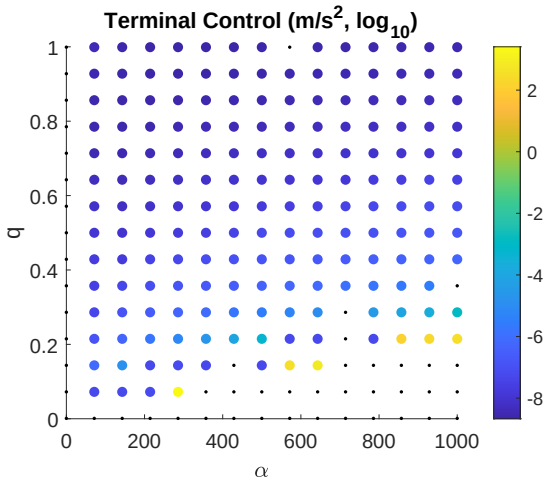

Fig. 7.40: The control magnitude at time $t = T_{max}$ as a function of the q and α parameters. Note that the magnitudes are presented as $log_{10}(x)$.

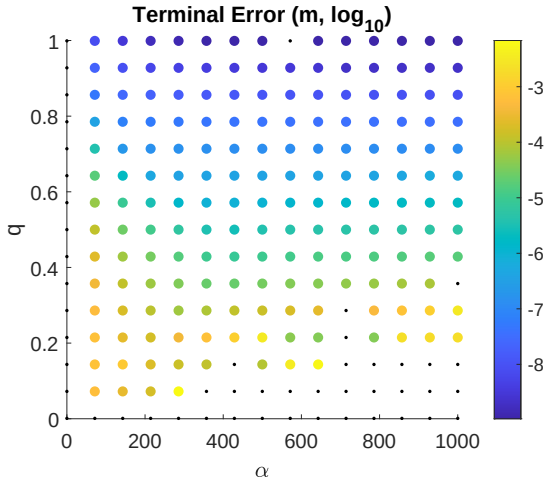

Fig. 7.41: The position error at time $t = T_{max}$ as a function of the q and α parameters. Note that the magnitudes are presented as $log_{10}(x)$.

7.4.4 Results

From these results, a selection is made of $q = .996$ and $\alpha = 300$ for the control law with the convergence bound set as $T_c = 300\,\text{s}$. The switching function was chosen to be $\tanh(x)$. Using these parameters, a set of 50 simulations are run with initial conditions that satisfy

$$|x_0| = \begin{bmatrix} 390\,\text{m} & 390\,\text{m} & 390\,\text{m} & 1.5\,\text{m/s} & 1.5\,\text{m/s} & 1.5\,\text{m/s} \end{bmatrix}^T \tag{7.38}$$

with simulations run using both the CW and two-body dynamics with the dual-input in-plane controls. The controls and dynamics were both updated at a rate of 20 Hz.

Unperturbed CW Dynamics

The position errors using the CW dynamics are shown in Figures 7.42 and 7.43 for the in-plane and cross-track subsystems, respectively. Both show two distinct phases. From the start of the simulation to about $t = 80\,\text{s}$, the error generally decreases smoothly for both subsystems, at which point the convergence rate sharply increases and the error drops rapidly. At the transition point, there is some brief transitory behavior in the errors, but overall the errors decrease smoothly.

The magnitudes of the errors, which drop below 10^{-80} m, are also worthy of note. Although the sample-and-hold approach to implementing the control laws has been seen to introduce disturbances, this control law is able to converge much closer than any of the other control laws.

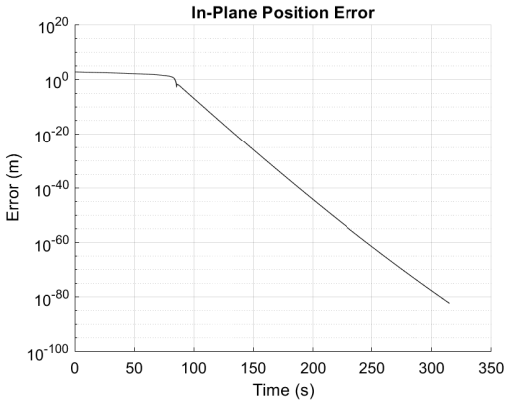

Fig. 7.42: In-plane position error for the unperturbed CW dynamics.

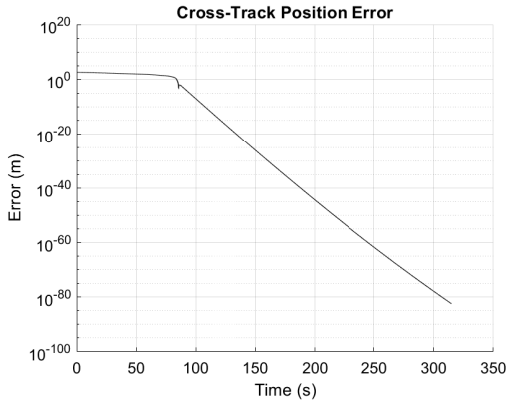

Fig. 7.43: Cross-track position error for the unperturbed CW dynamics.

Similar results are also seen in the control magnitude plots, with Figure 7.44 showing the in-plane control magnitudes and Figure 7.45 showing the cross-track control magnitudes. As with the error plots, there are two distinct phases: the initial control is relatively constant until $t = 80$ s, after which the control rapidly drops off. There are also brief periods of transitory periods that divide these two phases. At the beginning of the simulation, both subsystems experience some variance in the control magnitude. Then, when the control magnitude starts to sharply drop around the $t = 80$ s mark, there is a sudden, but momentary increase in the error before it begins descending again. Further analysis, not presented here, shows that these spikes result from using the hyperbolic tangent as a smoothing function and the height of the peaks changes with the γ scaling factor.

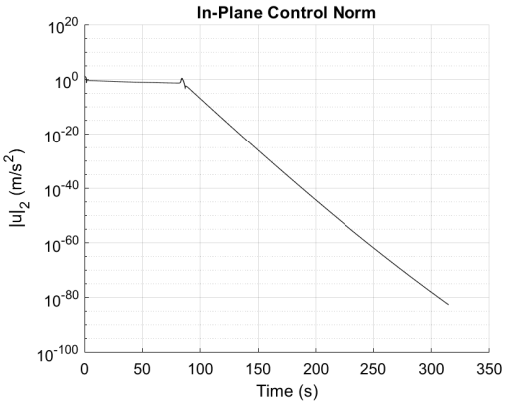

Fig. 7.44: In-plane control for the unperturbed CW dynamics.

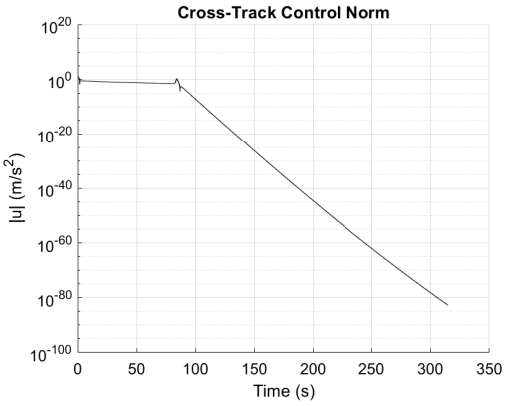

Fig. 7.45: Cross-track control for the unperturbed CW dynamics.

Two-Body Dynamics

The results for the two-body dynamics are now presented. In Figures 7.46 and 7.47, the position errors are shown for the in-plane and cross-track subsystems. The main difference seen from the CW results is that the perturbations in the system prevent the error from decreasing to same degree that they did in the CW dynamics, and some chattering is observable. Overall, the errors are on the order of 10^{-9} m for the in-plane subsystem and 10^{-10} m for the cross-track subsystem, which is tighter convergence than has been seen in the other control laws. The same sharp dip and rebound in the errors around $t = 80$ s is also evident. Due to the scaling of the Y-axis, the dip appears to be more prominent but is of similar magnitude to the CW results.

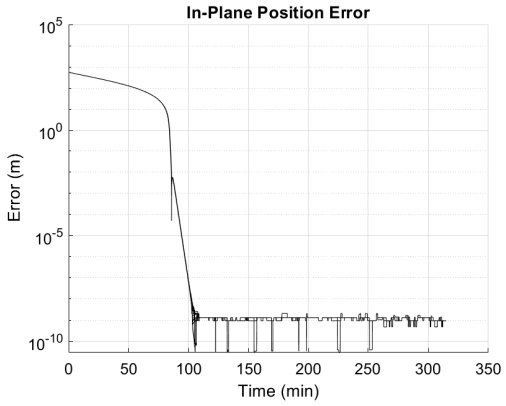

Fig. 7.46: In-plane position error for the two-body dynamics.

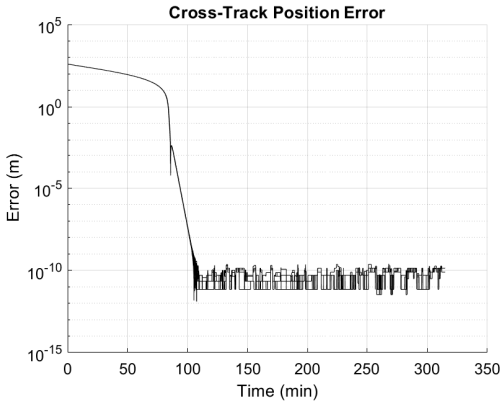

Fig. 7.47: Cross-track position error for the two-body dynamics.

The control plots are shown in Figures 7.48 and 7.49 for the in-plane and cross-track subsystems, respectively. Again, the perturbations result in the control magnitudes not decreasing to the extent seen in the CW results, but the control law still is only applying negligible levels of control. The control profiles still show some sudden changes in magnitude, but the overall impact is small. The peak in the control magnitude, around $t = 80\,\mathrm{s}$ is visible in these plots although, like the error plots, the magnitude is roughly the same as the CW results but more visible at this scale.

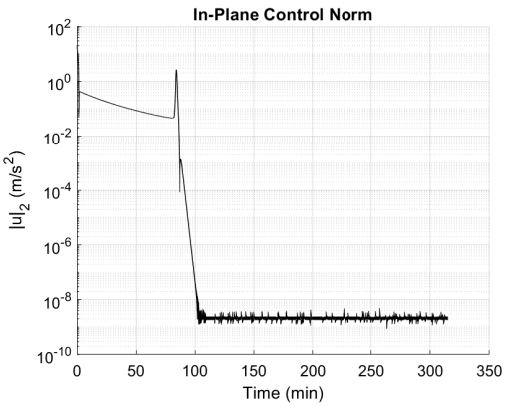

Fig. 7.48: In-plane control for the two-body dynamics.

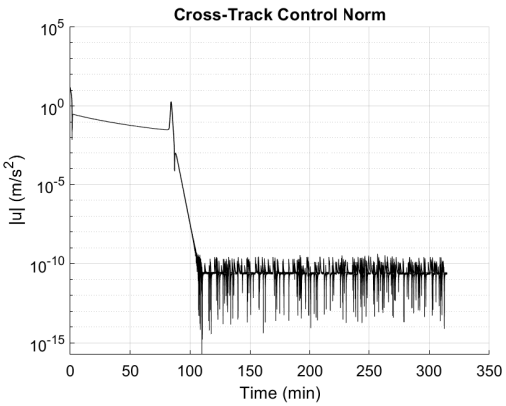

Fig. 7.49: Cross-track control for the two-body dynamics.

Overall, this control law is able to successfully stabilize the two-body system with J2 and drag perturbations. The use of the hyperbolic tangent function in place of the sign function for the switching condition actually seems to increase the range of conditions for which the system can be stabilized. However, there is no clear connection between the ω function used in the control law and the control behavior so tuning must be done through experimentation.

7.5 Conclusion

This chapter investigated three control laws which use various surfaces to define the behavior of the system. The switching surface control is based on the minimum-time solution where the control direction is defined by its state relative to the surface. The surfaces are optimized in order to minimize the maximum instantaneous control applied to the system. The control was successful in driving the system to a neighborhood of the origin but experienced chattering behavior.

The fractional power sliding mode control and backstepping sliding mode control attempt to hold the system on some surface with satisfactory, finite-time convergence properties. In both cases, the control laws were able to stabilize the system but experienced chatter. The use of the hyperbolic tangent to approximate the sign function was investigated and found to be successful at minimizing chatter when used with an appropriate scaling factor.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

The contribution of this dissertation was to provide strategies and guidance for implementing finite-time control laws for the spacecraft rendezvous problem. This was accomplished for multiple control laws with convergence times, state and control bounds, and overall performance being investigated using both simulations and analytic approaches. The controllers were applied to the spacecraft rendezvous scenario and tested using both the unperturbed, relative orbital dynamics and the two-body dynamics with J2 and drag perturbations. While the tuning parameters used in each controller are specific to the spacecraft rendezvous problem, the approaches, insights, and guidelines are applicable to any system using one of these controllers.

In Chapter 4, a guidance trajectory was generated and followed by a robust, tube-based MPC. A set of initial conditions were found for the guidance trajectory that guarantee it remained within the allowable state, control, and time constraints. The tube-based MPC used to follow this trajectory was developed using a feedback control and the expected perturbations to shrink the nominal state and control bounds. These reduced bounds guaranteed that perturbations cannot drive the system outside the nominal bounds.

Since the open-loop guidance trajectory was shown to be feasible over a fixed time horizon for all states in the initial set, it guaranteed fixed-time convergence for the open-loop, unperturbed system. The addition of the tube-based MPC did allow the system to reject perturbations, but it was only able to do so asymptotically. This highlights one point of potential nuance where a controller may guarantee finite-time stability for the unperturbed system but fail to achieve exact convergence to the desired state when perturbations are added. As with all trade-offs, this may be acceptable behavior or not, depending on the system requirements and desired level of performance.

A common Lyapunov control was implemented in Chapter 5. This control was obtained by finding a Lyapunov function that satisfied both the spacecraft rendezvous problem and a predetermined, reference system. At each point in time, the control for the system was then found as a function of the common Lyapunov function and guaranteed finite-time convergence.

Examining the three matrices used in the common Lyapunov function, the upper bounds on the convergence time and the maximum control were found to be functions of the eigenvalues of the matrices. This was utilized to influence the controller to have desirable properties by developing a semidefinite program that minimized the convergence time and maximum control, while still maintaining finite-time stability. The semidefinite program was successful in driving the bounds toward the desired values, but the bound on the convergence time was found to be very conservative and a poor estimate of the actual convergence time of the system. The maximum control bound did prove to be a reasonable estimate, especially for smaller bounds. However, the use of the Brunovsky transformation with the system dynamics meant that while the common Lyapunov control respected these bounds, the Brunovsky transformation could cause the control to exceed the bounds.

Since the in-plane and cross-track dynamics were independent, they each had separate controllers. Interestingly, the $\mathbb{R}^2$, cross-track dynamics consistently had less conservative bounds compared to the $\mathbb{R}^4$, in-plane dynamics. This suggests that the this method of tuning the common Lyapunov control is best suited to smaller dimensional systems. Additionally, both systems experienced chatter as the subsystems neared the origin.

A backstepping control was used in Chapter 6 for the rendezvous problem. This control relied on the use of a tuning function which had initial and final conditions for it and its derivatives. From these conditions, the tuning function was constructed using a linear combination of basis functions. The convergence time bound of the system was the time corresponding to the final condition on the tuning function so the system was predefined-time stable.

Additionally, the control is proportional to the highest derivative of the tuning function, so by optimizing that, the control applied to the system can be influenced. This was done by adding extra basis functions to the tuning function, beyond what was needed to satisfy the initial and final constraints. The tuning function was then optimized to minimize the maximum instantaneous control magnitude. This was shown to be successful for the Brunovsky system of chained integrators, but any improvements to the system were negated due to the conversion of the Brunovsky control back to the actual control.

In Chapter 7, three different controllers which used switching surfaces were examined. The first controller was developed based on the minimum time solution for the double integrator problem. By choosing a control magnitude, a surface along which the system will be driven to the origin was defined. A second control magnitude was then used to drive the system to the surface. While these control magnitudes were constant in the Brunovsky dynamics, the actual control applied to the system varied as the states changed. The Brunovsky magnitudes were then optimized to minimize the actual control being applied to the system.

This approach was successful in developing a finite-time control law that reduced the maximum control being applied to the system. However, the structure of the problem was very rigid with little to no ability to tune the controller.

The second two control laws were designed to hold the system on a switching surface which guaranteed predefined-time convergence. However, in the presence of perturbations, these controllers developed chatter as they failed to keep the system exactly on the surface. Additionally, while some tuning parameters were directly tied to the convergence bound and perturbation magnitude, additional parameters lacked clear connections to controller behavior.

The discontinuous sign function was replaced in both controllers with a continuous hyperbolic tangent approximation. This reduced the theoretical convergence to predefined-time convergence to a neighborhood of the origin. However, in practice, a properly scaled hyperbolic tangent function actually improved the performance of the controllers and allowed them to converge for a larger range of tuning parameters. These parameters were then tuned

using a grid search to find a controller which minimized the convergence time and maximum control usage.

From these results, the conclusions of this dissertation are as follows.

- Finite-time controllers prioritize convergence time, which complicate the processes of including or balancing other performance metrics.
- The Brunovsky transformation is useful in expanding the range of systems for which a control law can be applied. However, since the transformation includes a state feedback term, control bounds on the control law are no longer guaranteed.
- Bounds on the system performance for Common Lyapunov controllers tend to become less predictive and more conservative as the dimensionality of the system increases.
- The use of the hyperbolic tangent function as a smooth approximation to the sign switching function is shown to decrease the impact of perturbations arising from the discretization of the control law.
- The predefined-time backstepping approach is the most useful of the approaches examined in guaranteeing finite-time convergence since the convergence time appears as a tunable parameter, the bound is not overly conservative, the approach naturally minimizes chatter, and the tuning function can be shaped to minimize control for simple systems.

To address weaknesses identified, future work can focus on implementing these approaches on actual systems, expanding the range of dynamics the control laws can be applied to, and continuing to mitigate chatter. Since each controller was tested in simulation, future work which applies the approaches to physical systems could validate the results shown here. Additionally, in the region near the equilibrium point, where forces and controls are near zero, minute perturbations could change the behavior of the system.

While the Brunovsky transformation was successfully used to simplify the spacecraft rendezvous problem model dynamics, it did present some complications that could be addressed. For example, the use of the feedback term in the Brunovsky transformation means

that the controller is unable to enforce control bounds. However, as the feedback term is known, the overall control bounds could be integrated into the model dynamics by bounding the allowable states and shrinking the model control bounds. Alternatively, the range of dynamics applicable to a given control law could be expanded so that the Brunovsky transformation is no longer required. While the switching based control laws are designed around second-order systems and the common Lyapunov control assumes a system of chained integrators, they could be expanded to more general system dynamics which preclude the need to use the Brunovsky transformation.

Additionally, the rendezvous problem only considered a single target and chaser spacecraft. This could be expanded in the future to multi-spacecraft formations with distributed or cooperative control being used to achieve objectives at both the formation and individual spacecraft level.

While the chatter was mitigated for some controllers by applying a smoothing function, additional approaches and their effects can be studied. Techniques such as applying a low-pass filter to the control input, enforcing a minimum bound on the control magnitude, and changing the control update rate are potential approaches which were not explored in this work.

REFERENCES

[1] M. Ryu, S. Hong, and M. Saksena, "Streamlining real-time controller design: From performance specifications to end-to-end timing constraints," *Real-Time Technology and Applications - Proceedings*, pp. 91–99, 1997.

[2] E. Jimenez-Rodriguez, J. D. Sanchez-Torres, and A. G. Loukianov, "Backstepping design for the predefined-time stabilization of second-order systems," *2019 16th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE 2019*, 2019.

[3] B. Liu, M. Hou, C. Wu, W. Wang, Z. Wu, and B. Huang, "Predefined-time backstepping control for a nonlinear strict-feedback system," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 8, pp. 3354–3372, 2021.

[4] B. Liu, W. Wang, Y. Li, Y. Yi, and G. Xie, "Adaptive Quantized Predefined-Time Backstepping Control for Nonlinear Strict-Feedback Systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 9, pp. 3859–3863, 2022.

[5] Y. Hong, J. Wang, and D. Cheng, "Adaptive finite-time control of nonlinear systems with parametric uncertainty," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 858–862, 2006.

[6] Y. Hong, Z. P. Jiang, and G. Feng, "Finite-time input-to-state stability and applications to finite-time control," *SIAM Journal on Control and Optimization*, vol. 48, no. 7, 2010.

[7] F. Lopez-Ramirez, D. Efimov, A. Polyakov, and W. Perruquetti, "Finite-time and fixed-time input-to-state stability: Explicit and implicit approaches," *Systems and Control Letters*, vol. 144, p. 104775, 2020. [Online]. Available: https://doi.org/10.1016/j.sysconle.2020.104775

[8] X. He, X. Li, and S. Song, "Finite-time input-to-state stability of nonlinear impulsive systems," *Automatica*, vol. 135, 2022. [Online]. Available: https://doi.org/10.1016/j.automatica.2021.109994

[9] A. Fuller, "Relay control systems optimized for various performance criteria," *IFAC Proceedings Volumes*, vol. 1, no. 1, pp. 520–529, 1960. [Online]. Available: http://dx.doi.org/10.1016/S1474-6670(17)70097-3

[10] R. Bellman, I. Glicksberg, and O. Gross, "On the "bang-bang" control problem," *Quarterly of Applied Mathematics*, vol. 14, no. 1, pp. 11–18, 1956.

[11] M. Zhihong, A. P. Paplinski, and H. R. Wu, "A Robust Mimo Terminal Sliding Mode Control Scheme For Rigid Robotic Manipulators," *IEEE Transactions on Automatic Control*, vol. 39, no. 12, pp. 2464–2469, 1994.

[12] Y. Feng, X. Yu, and Z. Man, "Non-singular terminal sliding mode control of rigid manipulators," *Automatica*, vol. 38, no. 12, pp. 2159–2167, 2002.

[13] S. Yu, X. Yu, B. Shirinzadeh, and Z. Man, "Continuous finite-time control for robotic manipulators with terminal sliding mode," *Automatica*, vol. 41, no. 11, pp. 1957–1964, 2005.

[14] Y. Song, Y. Wang, J. Holloway, and M. Krstic, "Time-varying feedback for regulation of normal-form nonlinear systems in prescribed finite time," *Automatica*, vol. 83, pp. 243–251, 9 2017.

[15] Z. Kan, T. Yucelen, E. Doucette, and E. Pasiliao, "A finite-time consensus framework over time-varying graph topologies with temporal constraints," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 139, 7 2017.

[16] D. Tran and T. Yucelen, "Finite-time control of perturbed dynamical systems based on a generalized time transformation approach," *Systems and Control Letters*, vol. 136, 2 2020.

[17] Y. Song, H. Ye, and F. L. Lewis, "Prescribed-time control and its latest developments," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, pp. 4102–4116, 7 2023.

[18] T. Huang, J. Wang, H. Pan, and W. Sun, "Finite-Time Fault-Tolerant Integrated Motion Control for Autonomous Vehicles With Prescribed Performance," *IEEE Transactions on Transportation Electrification*, vol. 9, no. 3, pp. 4255–4265, 2023.

[19] H. Gao, Z. Kan, F. Chen, Z. Hao, X. He, H. Su, and K. Li, "Adaptive Finite-Time Trajectory Tracking Control of Autonomous Vehicles That Experience Disturbances and Actuator Saturation," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 2, pp. 80–91, 2022.

[20] Q. Hu, W. Chen, and L. Guo, "Fixed-Time Maneuver Control of Spacecraft Autonomous Rendezvous with a Free-Tumbling Target," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 2, pp. 562–577, 2019.

[21] F. Wang, Y. Miao, C. Li, and I. Hwang, "Attitude control of rigid spacecraft with predefined-time stability," *Journal of the Franklin Institute*, vol. 357, no. 7, pp. 4212–4221, 2020. [Online]. Available: https://doi.org/10.1016/j.jfranklin.2020.01.001

[22] Y. Zhang, R. Vepa, G. Li, and T. Zeng, "Mars Powered Descent Phase Guidance Design Based on Fixed-Time Stabilization Technique," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 4, pp. 2001–2011, 2019.

[23] C. Wu, J. Yan, J. Shen, X. Wu, and B. Xiao, "Predefined-Time Attitude Stabilization of Receiver Aircraft in Aerial Refueling," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 10, pp. 3321–3325, 2021.

[24] D. Wu, Y. Cheng, H. Du, W. Zhu, and M. Zhu, "Finite-time output feedback tracking control for a nonholonomic wheeled mobile robot," *Aerospace Science and Technology*, vol. 78, pp. 574–579, 2018.

[25] J. C. Schon and B. Andresen, "Finite-Time Optimization of Chemical Reactions: n A m B," *The Journal of Physical Chemistry*, vol. 100, no. 21, pp. 8843–8853, 1996.

[26] J. C. Schön, "Finite-time thermodynamics and the optimal control of chemical syntheses," *Zeitschrift fur Anorganische und Allgemeine Chemie*, vol. 635, no. 12, pp. 1794–1806, 2009.

[27] Y. Liu, H. Li, Z. Zuo, X. Li, and R. Lu, "An Overview of Finite/Fixed-Time Control and its Application in Engineering Systems," *IEEE/CAA Journal of Automatica Sinica*, vol. PP, no. 99, pp. 2106–2120, 2022.

[28] NASA, "Gemini Summary Conference Report," *Science and Technical Information Division, Office of Technology Utilization*, 1967. [Online]. Available: https://www.hq.nasa.gov/alsj/GeminiSummaryConf{_}SP-138.pdf

[29] E. E. Aldrin, "Line-of-Sight Guidance Techniques for Manned Orbital Rendezvous," Dissertation, Massachusetts Institute of Technology, 1963.

[30] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Santa Barbara, California: Nob Hill Publishing, 2019.

[31] I. Ananievskii, N. Anokhin, and A. Ovseevich, "Design of bounded feedback controls for linear dynamical systems by using common Lyapunov functions," *Theoretical and Applied Mechanics Letters*, vol. 1, no. 1, p. 013001, 2011. [Online]. Available: http://dx.doi.org/10.1063/2.1101301

[32] M. W. Harris and M. B. Rose, *Optimal Spacecraft Guidance*. Utah State University, 2023.

[33] V. G. Rao and D. S. Bernstein, "Naive control of the double integrator," *IEEE Control Systems*, vol. 21, no. 5, pp. 86–97, 2002.

[34] A. Polyakov, "Nonlinear feedback design for fixed-time stabilization of linear control systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 8, pp. 2106–2110, 2012.

[35] J. Akagi, "Spacecraft nonlinear control strategies," https://github.com/JAkagi-USU/SpacecraftNonlinearControlStrategies.

[36] ——, "Jakagi-usu/spacecraftnonlinearcontrolstrategies: Dissertation code," https://zenodo.org/doi/10.5281/zenodo.13290701.

[37] ——, "Supplemental data for: Nonlinear control strategies for rendezvous and proximity operations," https://digitalcommons.usu.edu/all_datasets/233, 2024.

[38] H. K. Khalil, *Nonlinear Systems*, 3rd ed. New Jersey: Prentice Hall, 2002.

[39] E. Jiménez-Rodríguez, A. J. Muñoz-Vázquez, J. D. Sánchez-Torres, and A. G. Loukianov, "A Note on Predefined-Time Stability," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 520–525, 2018.

[40] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," 1994.

[41] D. Zhou, S. Sun, and K. L. Teo, "Guidance laws with finite time convergence," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 6, pp. 1838–1846, 2009.

[42] W. J. Rugh, *Linear system theory.* Prentice-Hall, Inc., 1996.

[43] J. P. Hespanha, *Linear systems theory.* Princeton University Press, 2018.

[44] P. J. Antsaklis and A. N. Michel, *Linear systems.* Springer, 1997, vol. 8.

[45] H. D. Curtis, *Orbital Mechanics for Engineering Students*, 2013.

[46] O. Montenbruck and E. Gill, *Satellite Orbits: Models, Methods and Applications.* Springer Berlin Heidelberg, 2000, vol. 134.

[47] NASA, "SCaN Glossary." [Online]. Available: https://www.nasa.gov/reference/scan-glossary/

[48] NASA-JPL, "Spherical Harmonic Representation of the Gravity Field Potential," Tech. Rep. 2. [Online]. Available: https://spsweb.fltops.jpl.nasa.gov/portaldataops/mpg/MPG{_}Docs/SourceDocs/gravity-SphericalHarmonics.pdf

[49] D. A. Vallado, *Fundamentals of astrodynamics and applications*, 2nd ed. Microcosm Press, 2001.

[50] U. S. Atmosphere, "National oceanic and atmospheric administration," *National Aeronautics and Space Administration, United States Air Force, Washington, DC*, 1976.

[51] CalPoly, "Cubesat Design Specification Rev. 14.1," Tech. Rep., 2022. [Online]. Available: http://www.cubesat.org/images/developers/cds{_}rev12.pdf{%}5Cnhttp://scholar.google.com/scholar?hl=en{&}btnG=Search{&}q=intitle:CubeSat+Design+Specification{#}0

[52] T. Villela, C. A. Costa, A. M. Brandão, F. T. Bueno, and R. Leonardi, "Towards the thousandth CubeSat: A statistical overview," *International Journal of Aerospace Engineering*, vol. 2019, 2019.

[53] T. K. Smith, Z. Lewis, K. Olsen, A. M. Bulcher, and S. A. Whitmore, "A Miniaturized Green End-Burning Hybrid Propulsion System for CubeSats," *AIAA Small Satellite Conference*, pp. 1–12, 2020.

[54] H. Ford, L. R. Hunt, and R. Su, "A simple algorithm for computing canonical forms," *Computers and Mathematics with Applications*, vol. 10, no. 4-5, pp. 315–326, 1984.

[55] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, "Robust tube-based MPC for tracking of constrained linear systems with additive disturbances," *Journal of Process Control*, vol. 20, no. 3, pp. 248–260, 2010.

[56] R. Gonzalez, M. Fiacchini, T. Alamo, J. L. Guzman, and F. Rodriguez, "Online robust tube-based MPC for time-varying systems: A practical approach," *International Journal of Control*, vol. 84, no. 6, pp. 1157–1170, 2011.

[57] K. Dong, J. Luo, Z. Dang, and L. Wei, "Tube-based robust output feedback model predictive control for autonomous rendezvous and docking with a tumbling target," *Advances in Space Research*, vol. 65, no. 4, pp. 1158–1181, feb 2020.

[58] C. Buckner and R. Lampariello, "Tube-based Model Predictive Control for the Approach Maneuver of a Spacecraft to a free-tumbling Target Satellite," in *Annual American Control Conference (ACC)*. IEEE, 2018.

[59] S. V. Raković, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, "Invariant approximations of the minimal robust positively invariant set," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 406–410, 2005.

[60] M. Tillerson and J. P. How, "Advanced Guidance Algorithms for Spacecraft Formation-Keeping," *Proceedings of the American Control Conference*, vol. 4, pp. 2830–2835, 2002.

[61] T. K. Smith, J. Akagi, and G. Droge, "Model Predictive Control Switching Strategy For Safe Small Satellite Cluster Formation Flying," *Journal of Aerospace Systems*, 2023. [Online]. Available: https://doi.org/10.1007/s42401-023-00237-2

[62] T. Smith, J. Akagi, and G. Droge, "Satellite cluster flight using guidance trajectory and model predictive control," *The Journal of the Astronautical Sciences*, vol. 71, no. 2, p. 14, 2024.

[63] ——, "Model Predictive Control for Formation Flying Based on D'Amico Relative Orbital Elements," *Astrodynamics*, Accepted.

[64] S. P. Hughes, "General mission analysis tool (gmat)," Tech. Rep., 2016.

[65] R. N. Shorten and K. S. Narendra, "On the stability and existence of common Lyapunov functions for stable linear switching systems," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 4, 1998, pp. 3723–3724.

[66] R. Shorten, K. S. Narendra, and O. Mason, "A result on common quadratic Lyapunov functions," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 110–113, jan 2003.

[67] K. S. Narendra and J. Balakrishnan, "A Common Lyapunov Function For Stable Lti Systems With Commuting AMatrices," *IEEE Transactions on Automatic Control*, vol. 39, no. 12, pp. 2469–2471, 1994.

[68] D. Liberzon and R. Tempo, "Common Lyapunov functions and gradient algorithms," *IEEE Transactions on Automatic Control*, vol. 49, no. 6, pp. 990–994, jun 2004.

[69] E. K. Chong and S. H. Zak, *An introduction to optimization*. John Wiley \& Sons, 2013.

[70] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, pp. 284–289, 2004.

[71] K. C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3 - a MATLAB software package for semidefinite programming, version 1.3," *Optimization Methods and Software*, vol. 11, no. 1, pp. 545–581, 1999.

[72] R. H. Tütüncü, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Mathematical Programming, Series B*, vol. 95, no. 2, pp. 189–217, 2003.

[73] J. D. Sánchez-Torres, D. Gómez-Gutiérrez, E. López, and A. G. Loukianov, "A class of predefined-time stable dynamical systems," *IMA Journal of Mathematical Control and Information*, vol. 35, no. 1, pp. I1–I29, 2018.

[74] J. D. Sanchez-Torres, M. Defoort, and A. J. Munoz-Vazquez, "A Second Order Sliding Mode Controller with Predefined-Time Convergence," *2018 15th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE 2018*, vol. 2, no. 1, pp. 18–21, 2018.

[75] E. Jiménez-rodríguez, A. J. Muñoz-vázquez, J. D. Sánchez-torres, M. Defoort, and A. G. Loukianov, "A Lyapunov-Like Characterization of Predefined-Time Stability," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4922–4927, 2020.

[76] A. K. Pal, S. Kamal, S. K. Nagar, B. Bandyopadhyay, and L. Fridman, "Design of controllers with arbitrary convergence time," *Automatica*, vol. 122, 2020.

[77] E. Jiménez-Rodríguez, J. D. Sanchez-Torres, and A. G. Loukianov, "On Optimal Predefined-Time Stabilization On Optimal Predefined-Time Stabilization," *International Journal of Robust and Nonlinear Control*, 2017.

[78] J. Lv, X. Ju, and C. Wang, "Neural Network-Based Nonconservative Predefined-Time Backstepping Control for Uncertain Strict-Feedback Nonlinear Systems," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2023.

[79] Y. Zhang, B. Niu, X. Zhao, P. Duan, H. Wang, and B. Gao, "Global Predefined-Time Adaptive Neural Network Control for Disturbed Pure-Feedback Nonlinear Systems With Zero Tracking Error," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 6328–6338, 2023.

[80] E. W. Weisstein, "Legendre Polynomial." [Online]. Available: https://mathworld.wolfram.com/LegendrePolynomial.html

[81] J. P. LaSalle, "Time Optimal Control Systems," *Proceedings of the National Academy of Sciences*, vol. 45, no. 4, pp. 573–577, 1959.

[82] D. W. Bushaw, "Differential equations with a discontinuous forcing term," Ph.D. dissertation, Princeton University, 1952.

[83] C. M. Kellett and P. Braun, *Introduction to nonlinear control: Stability, control design, and estimation*. Princeton University Press, 2023.

[84] B. R. Barmish and G. Leitmann, "On Ultimate Boundedness Control of Uncertain Systems in the Absence of Matching Assumptions," *IEEE Transactions on Automatic Control*, vol. 27, no. 1, pp. 153–158, 1982.

[85] F. Blanchini, "Ultimate Boundedness Control for Uncertain Discrete-Time Systems via Set-Induced Lyapunov Functions," *IEEE Transactions on Automatic Control*, vol. 39, no. 2, pp. 428–433, 1994.

[86] A. Levant, "Principles of 2-sliding mode design," *Automatica*, vol. 43, no. 4, pp. 576–586, 2007.

CURRICULUM VITAE

John Tamotsu Akagi

Education

- MS, Aerospace Engineering, Utah State University, 2022
- BS, Mechanical Engineering, Brigham Young University, 2019

Published Journal Articles

- Smith, Tyson, John Akagi, and Greg Droge. "Model Predictive Control for Formation Flying Based on D'Amico Relative Orbital Elements." Astrodynamics (Accepted)
- Smith, Tyson, John Akagi, and Greg Droge. "Satellite Cluster Flight Using Guidance Trajectory and Model Predictive Control." The Journal of the Astronautical Sciences 71.2 (2024): 14.
- Smith, Tyson, John Akagi, and Greg Droge. "Model predictive control switching strategy for safe small satellite cluster formation flight." Aerospace Systems 6.4 (2023): 559-579.
- Akagi, John, T. Devon Morris, Brady Moon, Xingguang Chen, and Cameron K. Peterson. "Gesture commands for controlling high-level UAV behavior." SN Applied Sciences 3, no. 6 (2021): 603.

Published Conference Papers

- Smith, Tyson, John Akagi, and Greg Droge. "Spacecraft Formation Flying Control Switching Surface Based on Relative Orbital Elements" 2022 AAS/AIAA Astrodynamics Specialist Conference

- Akagi, John, Randall S. Christensen, and Matthew W. Harris. "Centralized uav swarm formation estimation with relative bearing measurements and unreliable gps." 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS). IEEE, 2020.

- Akagi, John, Brady Moon, Xingguang Chen, and Cameron K. Peterson. "Gesture Commands for Controlling High-Level UAV Behavior." In 2019 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1023-1030. IEEE, 2019.

Patent Applications

- "Spacecraft Formation Control for Fixed Thrust Propulsion", U.S. Provisional Pat. Application No. 18/611,850, filed March 21, 2024

- "Maneuvering A Spacecraft Using an Optimized Guidance Trajectory and Model Predictive Control", U.S. Provisional Pat. Application No. 18/528,385, filed December 4, 2023

- "Maneuvering A Spacecraft Using an Optimized Guidance Trajectory and Model Predictive Control", U.S. Provisional Pat. Application No. 18/465,505, filed September 12, 2023

- "Model Predictive Control for Spacecraft Formations", U.S. Provisional Pat. Application No. 17/891,968, filed August 19, 2022

- "Model Predictive Control for Spacecraft Formations", U.S. Provisional Pat. Application No. 17/689,038, filed March 8, 2022