LOW-THRUST CONTROLLER FOR SLOT-BASED SATELLITE CONSTELLATIONS

by

Mario Melendrez Contreras

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author . . . .
Department of Aeronautics and Astronautics
May 18, 2021

Certified by . . . .
Richard Linares
Charles Stark Draper Assistant Professor
Thesis Supervisor

Accepted by . . . .
Zoltan Spakovszky
Professor, Aeronautics and Astronautics
Chair, Graduate Program Committee

LOW-THRUST CONTROLLER FOR SLOT-BASED SATELLITE CONSTELLATIONS

by

Mario Melendrez Contreras

Submitted to the Department of Aeronautics and Astronautics
on May 18, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Due to the increase in popularity of satellite constellations, some altitudes in LEO have experienced a large increase in number of satellites. This is a trend expected to continue in the future, which could potentially lead to an increased risk of collision between satellites. Collision avoidance is therefore paramount to maintain normal operations and to prevent runaway growth of space debris in LEO. To that end, this thesis develops state-space LQR and tube MPC controllers for LEO satellites operating in near-circular orbits with low-thrust engines. This is done using a linearized model of the dynamics under the Earth gravitational potential and the atmospheric drag.

Thesis Supervisor: Richard Linares
Title: Charles Stark Draper Assistant Professor

Acknowledgments

I would first like to acknowledge my family who has made many sacrifices throughout their life to give me the best opportunities to pursue a good education. Thank you and I will never cease to cherish your love and support. I would also like to thank David Arnas for all his help and encouragement in tackling this problem.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As many in the aerospace field know, satellite constellations [5,6,8,12,22,26,31,38] are quickly becoming a very popular space architecture for space missions. This is due to the numerous benefits that they provide when compared to individual spacecrafts. In general, satellite constellations are able to improve the response of the system while providing more optimal and lower cost solutions to the mission goals. For example, the ESA's Sentinel-1 mission [33] consists of a two satellites aimed at providing data continuity for the Sentinel program by using a configuration that optimizes coverage of the Earth. Additionally, there are some situations, for instance, in global coverage missions, that can only be achieved by large satellite constellations. Two examples of this kind of mission are SpaceX's Starlink [35] and Amazon's Project Kuiper [13], which aim to provide constant and global internet coverage. Other situations in which constellations have proven advantageous are when coordinated measurements can produce more useful data than when compared with sole satellites. An example of this is NASA's A-Train constellation [36], which consists of multiple satellites taking measurements over the same Earth regions, which improves our knowledge of Earth's environment and climate. However, with the rise in popularity comes the increased risk of collisions between satellites or satellite constellations. There is also a danger that satellites not operating with adequate collision avoidance will result in a future where orbits are dictated by random collisions with debris rather than with controlled satellites [14]. Collision avoidance is therefore paramount in order to ensure the safety of current and future missions in the Low Earth Orbit (LEO) region.

In that regard, extensive research has been performed by a large number of authors in the field of collision avoidance [17, 21, 29] and orbit maintenance [4, 19, 37] for satellite constellations. Research performed in that direction includes the assessment of the probability of collision [27, 28], autonomous control [40], the definition of a complete slotting architecture for the LEO region [7], the minimization of the ΔV needed to maximize the likelihood that a collision would be avoided [10, 34], or the use of differential drag to perform the control maneuvers of the formation [18, 24, 32]. Related to this, it is the control strategies used in formation flying. Examples of that include the helix formation of the TandemX mission [15], or the control box strategy [1, 9] for Flex mission [25].

This work deals with the generation and study of control law strategies for LEO satellites compliant with a constellation slotting architecture as the one proposed in Ref. [7]. In these kind of architectures, satellites are deployed in slots, which are zones in which only one satellite is allowed to operate in. These slots assure that there will be no conjunction between slots under a normal operation of the architecture. Therefore, ensuring that satellites stay within their slots ensures that no conjunctions will occur. In particular, this thesis focuses on the application of different control laws to deal with satellite maneuvers while maintaining the safety of the constellation as a whole. This means that the control system proposed in this thesis assures a minimum distance between the satellites of the constellation even under a worst case scenario where a spacecraft must perform a collision avoidance maneuver, as long as that maneuver stays within the slot. To that end, a control scheme is proposed based on Linear Quadratic Regulator (LQR) techniques [16] applied to a linearized model of the dynamics of a satellite located in LEO under the perturbation of the Earth's gravitational potential and the atmospheric drag [2]. A tube model predictive controller (MPC) based on bounded disturbances, state constraints, and control constraints is also applied [23]. In particular, for both controllers, a state-space system is created using the position of a satellite in altitude and along-track distance as states. This control scheme assumes that the satellites are using low thrust engines in a near-circular Low Earth Orbit and that continuous information about the satellites' states is available. The main advantage of the proposed control scheme is that it is specially suitable for real time on board control of spacecraft due to its very low computational requirements. In addition, the tube MPC also has the benefit of ensuring the satellite does not exit the slot at the expense of less slot area. By ensuring the satellite does not exit the slot, the tube MPC guarantees that collisions between satellites in a slot-based constellation will not occur.

The work in this thesis is organized as follows. First, the methods used to generate a linear, simplified model of a satellite subjected to the perturbation produced by the Earth gravitational potential and the atmospheric drag while operating within a slot in Low Earth Orbit is described. A summary of the LQR and tube MPC methods for determining the controllers is also discussed. Following this, the results are presented for an example of a LEO satellite maintaining its reference altitude, implementing an altitude change, and trying to follow along-track commands using both the LQR and tube MPC control schemes. A discussion of the advantages and limitations of these controllers is included afterwards. Finally, a conclusion section summarizing the work and discussing possible improvements is presented.[1]

[1] Much of the work regarding the introduction to this problem and the development, results, and discussion surrounding the LQR controllers presented here is adapted from a manuscript from the author. This manuscript can be found in Ref. [11]

Chapter 2

Methods

2.1 Linearized Model

In order to create a state-space model of the dynamics of a satellite subjected to orbital perturbations, a linear model is required. To that end, we make use of the approximated analytical model from Ref. [2] to simplify the system of differential equations obtained from the orbital perturbations. In this model, it is assumed that the satellites are in near-circular orbits and subjected to the effects of the perturbation produced by the Earth gravitational potential and the atmospheric drag. Other perturbations such as the solar radiation pressure or the third body are considered negligible compared to the considered perturbations. Therefore, following Ref. [2] we can derive that the evolution of the semi-major axis of the orbit can be approximated by

$$\dot{a} = -\rho \frac{S}{m} C_d \sqrt{\mu} a^{\frac{1}{2}} + \frac{2}{\sqrt{\mu}} a^{\frac{3}{2}} T_\theta \tag{2.1}$$

where a is the semi-major axis of the satellite, ρ is the atmospheric density, S is the cross-sectional area of the satellite, m is the mass of the satellite, C_d is the drag coefficient of the satellite, μ is the Earth's gravitational constant, T_θ is the acceleration produced by the thrusters in the direction opposite to the motion, and $\dot{a}$ is the rate of change of the semi-major axis with respect to time. Note that it is assumed that the variation in the semi-major axis is small compared to its value, and that the atmospheric density experienced by the satellite is not affected by these small changes in its altitude. On the other hand, the equation for the motion of the along-track time delay between the satellite and its nominal position is

$$\dot{\Delta T} = -\frac{3}{2}\frac{(a - a_0)}{a_0} \tag{2.2}$$

where $\dot{\Delta T}$ is the rate of change of along-track time delay and a_0 is the nominal semi-major axis of the satellite orbit, that is, the semi-major axis defined under the effects of the perturbation produced by the Earth gravitational potential [2,3]. To simplify the notation used in this thesis, k_1 and k_2 are defined as

$$k_1 = -\rho\frac{S}{m}C_d\sqrt{\mu}; \qquad k_2 = \frac{2}{\sqrt{\mu}}. \tag{2.3}$$

Since the state-space model for control requires the dynamic equations to be completely linear, a Taylor series expansion is performed on Equation 2.1 to linearize it around the operating point of a_0, while retaining the first order terms of the expansion. This results in a linearized equation for the rate of altitude change given by

$$\dot{a}_{\text{linear}} = k_1 a_0^{\frac{1}{2}} + k_2 a_0^{\frac{3}{2}} T_\theta + \left(\frac{1}{2}k_1 a_0^{-\frac{1}{2}} + \frac{3}{2}k_2 a_0^{-\frac{1}{2}} T_\theta\right)(a - a_0) \tag{2.4}$$

In that regard, and in order to check the precision of the linearization performed in Equation 2.4, a series of numerical comparisons are performed with respect to the analytical solution from Equation 2.1. In particular, Figure 2-1 shows the result of this comparison for an orbit with a nominal altitude of 705 km. The parameters used in this equation are given in Table 2.1. As can be seen, the linearization has a percentage error of less than 0.008% for an altitude deviation of 105 km. This altitude deviation is more than 3 orders of magnitude higher than what is considered in this problem and as a result the linearization performed is valid for the problem that is considered here.

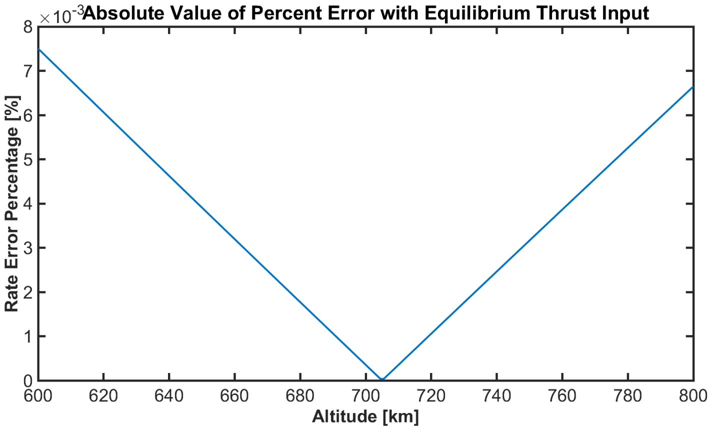

Figure 2-1: Linearization error

Table 2.1: Satellite and orbit parameters

Variable	Name	Value	Unit
ρ	Atmospheric density	10^{-14}	kg/m^3
S	Cross-sectional area	8.5	m^2
m	Mass	1000	kg
C_d	Coefficient of drag	2.2	-
μ	Earth gravitational constant	3.98600441188*10^{-14}	m$^3/s^2$
a	Semi-major axis	7076*10^3	m

2.2 State-Space Formulation

The goal of this section is to generate a state-space representation of the dynamics presented in the previous subsection in order to apply LQR and tube MPC techniques to control the altitude and along-track states. In order to accomplish this, the states of the system and their derivatives are first defined. Afterwards, the state-space model is centered around the operating point of the nominal semi-major axis and position of the satellite. Then, the controllability and observability matrices are determined to ensure that the system is controllable and observable for the parameters chosen.

In the state-space formulation, a control matrix K governs the closed-loop eigenvalues of the system, meaning that the control matrix will govern the response of the system. To create a state-space system, the states and their derivatives are defined as

$$x = \begin{bmatrix} a \\ \Delta T \end{bmatrix}; \qquad \dot{x} = \begin{bmatrix} \dot{a} \\ \Delta \dot{T} \end{bmatrix}. \tag{2.5}$$

This results in a state-space formulation where the derivative of the state vector can be described as

$$\dot{x} = Ax + Bu \tag{2.6}$$

where A and B are matrices that relate the effects of the states and control inputs to the future states respectively. We can also describe an output Y that is defined as

$$Y = Cx + Du \tag{2.7}$$

where C and D are matrices that describe the effects of the states and control inputs on the output. In general, the output Y simply tracks the states we are interested in. In order to create a state-space formulation centered around the operating point of a_0, it is necessary to obtain the equilibrium thrust that makes $\dot{a}_{\text{linear}}$ equal to zero. This results in an equilibrium thrust of

$$T_{\theta eq} = \frac{-k_1}{k_2 a_0} \tag{2.8}$$

which represents the thrust required to cancel out the drag loss of the satellite. The A and B matrices necessary for the state-space formulation are found by taking the partial derivatives of Equation 2.2 and Equation 2.4 with respect to the states and plugging in the equilibrium thrust. This results in the following A and B matrices:

$$A = \begin{bmatrix} \frac{-k_1}{2\sqrt{a_0}} & 0 \\ \frac{-3}{2a_0} & 0 \end{bmatrix}; \qquad B = \begin{bmatrix} k_2 a_0^{\frac{3}{2}} \\ 0 \end{bmatrix}. \tag{2.9}$$

As mentioned above, the C matrix relates the effects of the states to the output. For the problem considered in this thesis we are simply interested in determining the states themselves as the output, so it is assumed that both states are directly measured. Therefore, the C matrix is set to $\begin{bmatrix} 1 & 1 \end{bmatrix}$. The D matrix is often zero for many physical systems. For this controller formulation, we are interested in the state variables as our output, so it is possible to simply set the D matrix to zero. In other complex systems this is not possible and the D matrix must be non-zero to define the output. The complete state-space formulation therefore is defined as

$$\begin{bmatrix} \dot{a} \\ \dot{\Delta T} \end{bmatrix} = \begin{bmatrix} \frac{-k_1}{2\sqrt{a_0}} & 0 \\ \frac{-3}{2a_0} & 0 \end{bmatrix} \begin{bmatrix} a \\ \Delta T \end{bmatrix} + \begin{bmatrix} k_2 a_0^{\frac{3}{2}} \\ 0 \end{bmatrix} T_\theta; \tag{2.10}$$

$$Y = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ \Delta T \end{bmatrix}. \tag{2.11}$$

It is important to note that this state-space formulation is only related to the difference in value from the operating point. Any change in semi-major axis is really a change from the nominal semi-major axis a_0. The same idea is applied to the along-track and to the thrust. To recoup these values one must simply add the deviation to the nominal operating point values.

2.2.1 Controllability and Observability of State-Space System

The controllability and observability matrices are important in modern control theory. The controllability matrix, if full rank, guarantees that it is possible to transfer our system from any initial state to any final state in a finite period of time. In a similar way, the observability matrix, if full rank, guarantees that we are able to determine the values of our states at any point in time. Without this guarantee, it is possible that certain satellite states would not available or observable, making control an irrelevant question. To ensure that the system is controllable and observable the controllability and observability matrices are checked to ensure they are full rank. The controllability matrix is defined as $M_c = \begin{bmatrix} B & AB \end{bmatrix}$, and the observability matrix is defined as $M_o = \begin{bmatrix} C & CA \end{bmatrix}^T$. Both of these matrices are full rank in this formulation. This is an important conclusion because in the Results section of this thesis it is shown that certain controllers do not possess the ability to reach certain states in a timely manner or at all. This is not due to the inherent dynamics of the problem because the observability and controllability matrices are full rank. Instead, as is shown later, the controllers are not able to reach these states due to other limitations which are discussed.

Chapter 3

Control

3.1 LQR Methods

Once the controllability and observability have been ensured to be full rank, the control matrix K is calculated. This is done using the LQR technique. In this technique, the control matrix is generated as an optimization problem. We aim to produce the best performance with as little cost as possible. Weights are applied to states, such as the semi-major axis and along-track delay, and control inputs. The relative weight between these variables determines the control matrix. The Q matrix represents the weights applied to the states. If the Q matrix is larger, the controller will work harder to fix the state errors. The R matrix represents the weight applied to the control input. If the R matrix is larger, the controller will minimize control input at the expense of state error. Bryson's Rule is used to determine the weights of the Q and R matrices. Bryson's Rule is a common first-pass method to determine the weights of an LQR controller. This method results in choosing the Q and R matrices as

$$Q = \begin{bmatrix} \dfrac{\alpha_1^2}{(x_{1max})^2} & 0 \\ 0 & \dfrac{\alpha_2^2}{(x_{2max})^2} \end{bmatrix}; \qquad R = \left[\psi \left(\frac{\beta_1^2}{u_{1max}^2} \right) \right]; \tag{3.1}$$

where x_{1max} and x_{2max} refer to the maximum semi-major axis and along-track delay states that are expected while u_{1max} represents the maximum control input that the thrusters are able to produce. α_1 and α_2 represent the relative weights on the states. β_1 represents the weight on the control input and ψ represents the weight between the states and the control input. The maximum control input, u_{1max}, varies depending on the engine so a variety of thrusts for low-thrust engines are collected from Ref. [39]. The thrust is divided by the satellite mass to determine the acceleration of the engines since Equation 2.1 is written in terms of engine acceleration and not engine thrust. Table 3.1 and Table 3.2 show thrust values for bi-propellant and ion engines.[1]

Multiple controllers with varying performance can be designed by adjusting the weights in Equation 3.1. This thesis explored two main controller weightings: altitude focused and along-track focused. These two controllers were both fed two different reference signals to track: the altitude and the along-track. These different controllers and their references are discussed in the Results section. The weighting values for these controllers are shown in Table 3.3, along with the values for x_{1max} and x_{2max}.

Table 3.1: Thrust values for bipropellant engines

Engine	Thrust (Newtons)
10N Biprop Thruster	10
5LbCb AMPAC	22
Leros	9
Monarc-5	4.5
Monarc-90	90
DOT-5	5
MR-103G	1.13
MR-11C	5.3
Solenoid Actuated 58-118 Thruster	3.5

Table 3.2: Thrust values for ion engines

Engine	Thrust (milliNewtons)
XIPS-13	17.2
XIPS-25	80-166
NSTAR	< 92
NEXFS	< 500
NEXT	< 236
T-6	145
RIT-XT	150
ETS-8	22

To determine the maximum values of the altitude, x_{1max}, and along-track, x_{2max}, we must first determine the size of the slot the satellite is expected to be within. To that end, we use the formulation proposed in Ref. [2] to relate the maximum variation

[1] The maximum value of the thrust range is used in the calculations for the results section.

Table 3.3: Bryson's rule values

Variable	Altitude	Along-Track	Unit
x_{1max}	112.75	112.75	m
x_{2max}	4.6635	4.6635	s
α_1	100	1	-
α_2	1	100	-
ψ	1	1	-
β_1	10	10	-

in the semi-major axis with the variation in along-track distance. This relationship is used to define a maximum slot boundary for a satellite in a constellation. This relation can be summarized as:

$$\Delta a = \frac{4}{\sqrt{3}}\sqrt{\frac{\Delta\lambda_s}{\omega_\oplus}\rho\frac{S}{m}c_d\sqrt{\mu a_0^3}} \tag{3.2}$$

where the variables $\Delta\lambda_s$ and $\omega_\oplus$ are the total along-track size in radians and the Earth spin rate in radians per second. First, from a mission analysis study of the worst case scenario in solar activity and taking into account the ballistic coefficient of satellites currently operating, we derive an along-track slot size of $\Delta L = 70$ km at 700 km of altitude imposing a maximum of 1 orbital maneuver each two weeks. This distance is converted to radians by dividing by the semi-major axis of the orbit, $\Delta\lambda_s = \Delta L/a_0$. This results in $\Delta\lambda_s = 0.009893$ radians or $0.5668°$. Introducing this result into Equation 3.2, results in a semi-major axis change of $\Delta a = 225.5$ m. The along-track size is also desired in time units. The equation

$$\Delta t = \frac{\Delta L}{v} = \Delta L\sqrt{\frac{a_0}{\mu}} \tag{3.3}$$

converts the along-track into time by using the orbital velocity. The along-track slot size in time is then $\Delta t = 9.327$ seconds. Our slot is defined to have this sizing for semi-major axis and along-track deviation. However, the linearization assumes that the satellite is in the center of this rectangle. Therefore, when x_{1max} and x_{2max} are defined, the values must be halved so that the total size of the slot is equal to the slot sizes, that is $x_{1max} = \Delta a/2$ and $x_{2max} = \Delta t/2$. Other satellite constellations, depending on their operating conditions, mission goals, satellite parameters, and other factors, will have to go through this calculation to determine their own slot dimensions.[1]

After determining the Q and R matrices, the control gain matrix K_{LQR} is found for each thruster case using MATLAB's lqr() command. A closed-loop controller was created using these K matrices and applied to the state-space system. The solution was propagated using a custom propagation program. In this program a saturation limit was implemented to ensure that the engine could only output the maximum thrust possible if the controller requests more thrust than is physically possible. The control action is calculated discretely so if the controller frequency is too low, the controller will output an outdated control input which has the ability to cause instabilities or overshoot. The control frequencies are included in the Results section.

3.2 Tube MPC Methods

The tube model predictive controller (MPC) is a type of controller that combines robustness in the form of a tube with a model predictive controller [23]. The model predictive controller performs optimization using a system model and can "look into the future" by performing the optimization using a finite horizon. How far it looks into the future can be tuned by adjusting how many steps it calculates. The tube portion of the tube MPC is a disturbance rejection controller that helps to constrain the states of the system within a bounded region or "tube".

The tube MPC takes into account the constraints of the system and actuators [23]. Because of the nature of the problem discussed here this makes the tube MPC a promising candidate for control. The linear system has constraints in the control action in that the thrusters cannot give infinite thrust. It has constraints in the states in that the satellite is not allowed to leave the slots. The tube MPC can also deal with bounded disturbances. In the problem proposed here, some error in the sensing of the state is expected. This sensor error can be implemented as a disturbance

[1] More information about slotting, satellite constellations proposals, and station-keeping can be found in Ref. [2,3,4,7].

on the state. It is important to note that any disturbance implemented within the tube MPC must be bounded to ensure the states do not exceed their limits. Once again, the maximum thrusts for the engines considered are provided in Table 3.1 and Table 3.2. The maximum states, x_{1max} and x_{2max}, are given in Table 3.3. In addition, the thruster is also assumed to have a disturbance. If the thruster is commanded to input a certain amount of acceleration, errors in the actuation system are assumed to change the actual control input seen by the satellite.

The tube is generated through an algorithm found in Ref. [30]. The algorithm uses the state and actuation constraints, as well as the uncertainty of the disturbance to calculate the tube. Because different engines have different thrust values some engines will be better at rejecting disturbances because they have sufficient thrust to do so. Some engines were able to reject up to a 5% error on the state whereas others were only able to reject up to a 1% error. This all depends on the thrust value, the specific state setpoint, and the assumed disturbance bounds. The uncertainty bounds for the state used for results will be noted since a uniform uncertainty percentage was not applied to all engines and all cases. The disturbance percentage on the control action will also be noted for the cases.

The disturbance rejection controller or tube also needs its own disturbance rejection gain. This was determined by using Bryson's rule as shown in Equation 3.1. The weighting values used are the same as the altitude controller shown in Table 3.3.

The MPC portion of the controller also needs a gain. The gains for both parts of the controller can be adjusted independently but in this case the tube and the MPC shared the same values. The MPC controller had a prediction horizon of $N =$ 10. Any results which differ in the prediction horizon will be explicitly called out. It is also important to note that the tube MPC is implemented discretely so the MATLAB command dlqr() was used to calculate the discrete gains. The controller is implemented at a frequency of 1 Hz unless otherwise specified. The code used to generate these results also makes use of the YALMIP optimization toolbox for MATLAB [20].

An example of the tube MPC is shown in Figure 3-1. The frequency of the controller and the prediction horizon are different in this example to more easily highlight its function. The outer box represents the limits of the states and the smaller boxes which encapsulate the state are the tubes. The state is guaranteed to land within the tube at any point in time despite uncertainties. The satellite is commanded to raise its altitude to the maximum state and the controller moves the state in that direction. However, once the outer-part of the tube lines up with the state limit, it no longer changes. This is because if the tube were to move more to the right then the state would not be guaranteed to stay within the slot. Once at this new location, the state moves around due to uncertainty in the state and disturbance with the control action. Despite these uncertainties, the state stays within the tube and ensures the satellite does not leave the slot.

Adjusting the uncertainty of the state will adjust the size of the tube. If the uncertainty is increased too much, the tube will grow to a point where many satellite maneuvers will not be possible since the tube will line up with the state limits. Larger uncertainties will limit the amount of states that are reachable within the greater slot. Smaller uncertainties will allow the satellite to reach more states. The gains on the disturbance rejection and the MPC will also affect this, as well as the total thrust available to the controller.

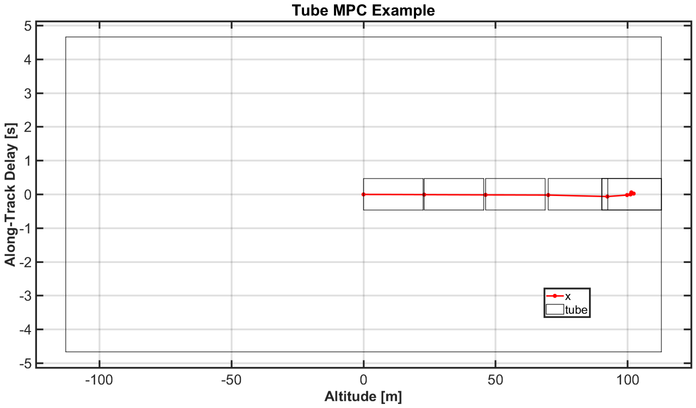

Figure 3-1: Tube MPC Example

Chapter 4

Results

4.1 LQR Methods

4.1.1 Altitude Controller

The altitude controller consists of an LQR controller that is weighted to penalize the semi-major axis state[1] 10 times more than the control input and 100 times more than the along-track state. The control thrusts generated from this controller are also limited by the maximum thrust that is physically possible to be generated from each engine. The controller is applied to both ion engines and bipropellant engines. Since the bipropellant engines have much higher thrusts when compared to the ion engines, the bipropellant engine controller updates its control input at a rate of 1 kHz in order to avoid instability. The ion engine controllers, on the other hand, are able to run only at 1 Hz to retain stability. The tests in this section are done to determine what a nominal collision avoidance maneuver looks like in this slot and determine its performance metrics in terms of error, settling time, and fuel costs. After these results, the specific use case of an example maneuver of a satellite trying to avoid a collision while remaining within the slot is explored.

Figure 4-1 depicts the LQR altitude tracking response of a x_{1max} (112.75 m)

[1] This is a notice to the reader that semi-major axis state and altitude state will be used interchangeably in this thesis.

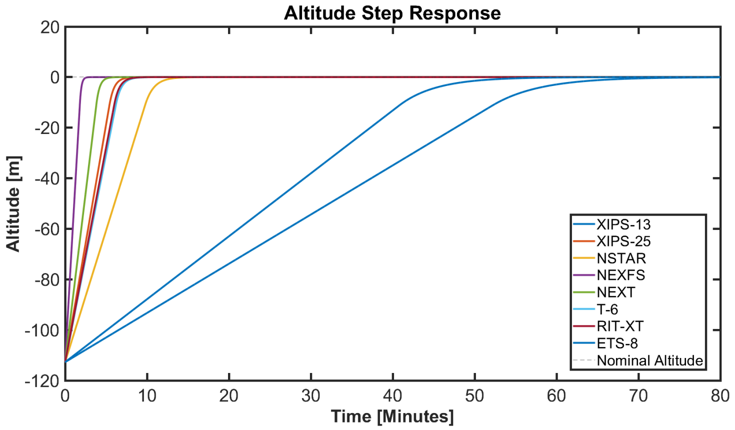

Figure 4-1: Altitude tracking with LQR control gains for ion engines

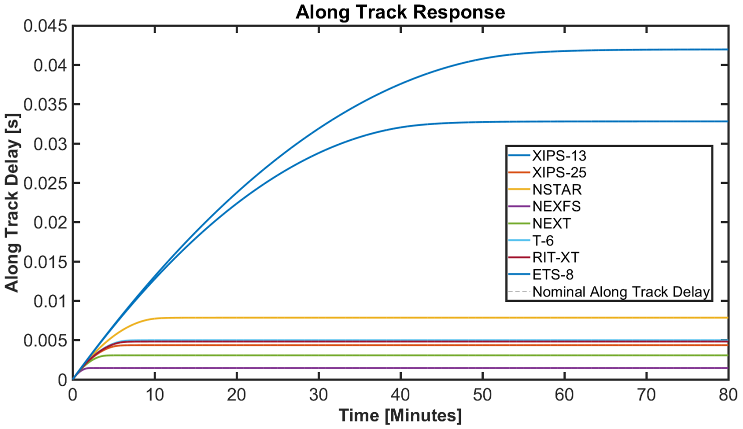

Figure 4-2: Along track delay response with LQR control gains for ion engines

step of satellites with different ion engines. The satellite parameters are provided in Table 2.1. This step size is chosen to demonstrate that the controller is able to handle large step inputs in the altitude state. The satellite starts at the bottom of the slot boundary $-x_{1max}$ to simulate an initial position after avoiding a collision. The satellite is commanded to upward to reach the desired steady-state semi-major axis position of zero. Once again, these values of altitude, along-track, and control input are differences from the equilibrium values used when generating the state-space system. Engines with a larger thrust value converged to the nominal altitude faster than those with a smaller value. Satellites with these higher thrust engines are also able to generate less error in the along-track delay because more thrust allows the satellite more quickly reach the setpoint altitude. This is seen in Figure 4-2.

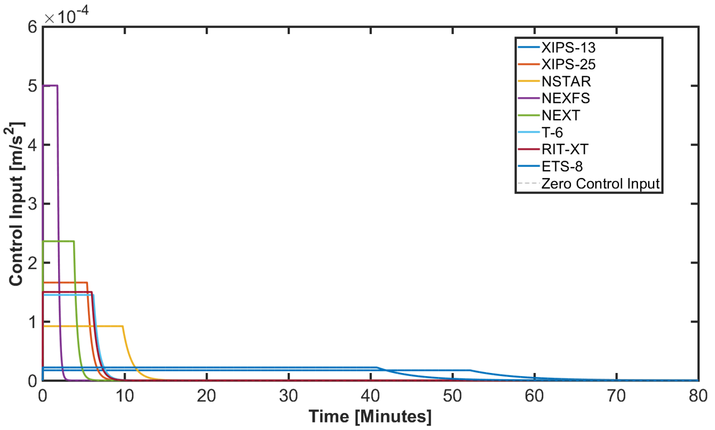

Figure 4-3: Control effort for LQR control gains for ion engines

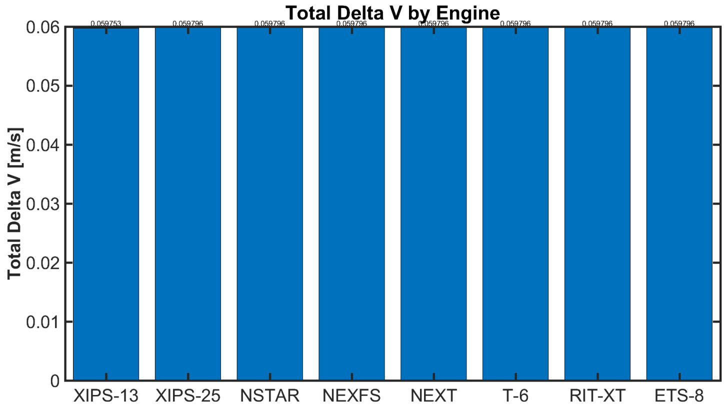

Figure 4-4: ΔV for ion engines

Figure 4-3 shows the control input needed to track the reference input. All the engines start at their maximum thrust and begin to throttle afterwards. The more the altitude state error is penalized, the longer the engines will function at maximum thrust. Engines with larger thrust operate at maximum thrust for less time since they provide greater amounts of acceleration, while weaker engines have to work at maximum thrust longer for the equivalent ΔV. The engine thrusts are tabulated in Table 3.1 for the bipropellant engines and Table 3.2 for the ion engines. NEXFS, the most powerful ion engine evaluated, takes about 3 minutes to reach the altitude reference using 500 mN of thrust. Figure 4-4 compares the ΔV of the maneuver, showing that these engines required the same ΔV. This is because none of the solutions overshoot the setpoint, so excess thrust is not used. These results show that a satellite with these parameters can use low-thrust ion engines to avoid collisions as long as there is advance notice and the trajectory within the slot is properly planned.

This x_{1max} (112.75 m) step was also applied to satellites using bipropellant engines. The thrust for these engines is much higher than that of the ion engines, as can be seen when comparing Table 3.1 and Table 3.2. Figure 4-5 shows the altitude tracking for the bipropellant engines. As a result of their higher thrust, the bi-propellant engines reach the nominal altitude much faster than their ion counterparts. Even the bipropellant engine with the least thrust, the MR-103G, reaches the nominal altitude in about 1 minute. The timescale on the along-track delay is similarly quick and is depicted in Figure 4-6. The error in the along-track delay is less than 7 milliseconds. Since these engines have much higher thrusts, they are operating at their maximum setpoint for a lower amount of time. This is shown in Figure 4-7. The ΔV is the same in this bipropellant case as the ion case. The ΔV for each engine is displayed in Figure 4-8. This is because none of the bipropellant controllers exhibit dynamics in which they overshoot the setpoint. If a satellite constellation is in need of faster responses these low-thrust bi-propellant engines are an option.

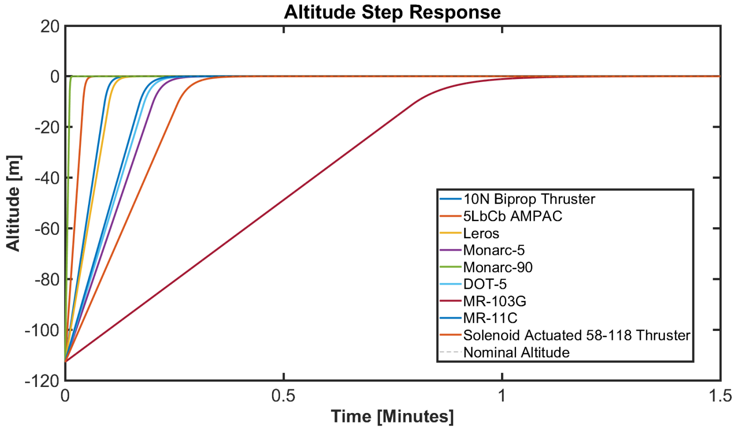

Figure 4-5: Altitude tracking with LQR control gains for bipropellant engines

Now we expand on the more specific situation of collision avoidance for the NEXFS engine. The NEXFS engine is selected for having the highest thrust in the class of ion engines examined here. We simulate a satellite in the nominal configuration raising its altitude to perform a collision avoidance maneuver. After being in the

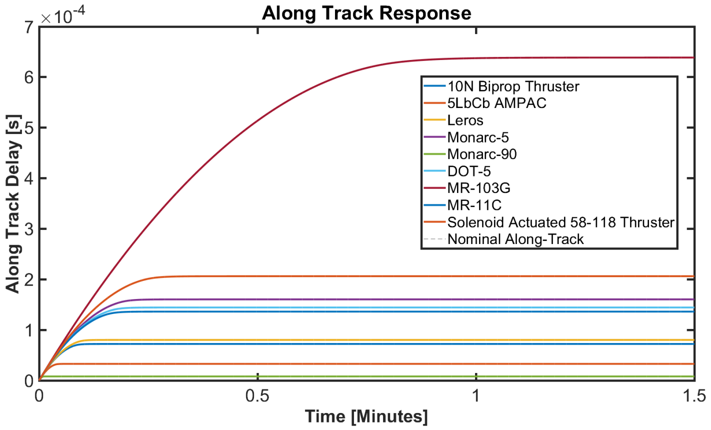

Figure 4-6: Along track delay response with LQR control gains for bipropellant engines

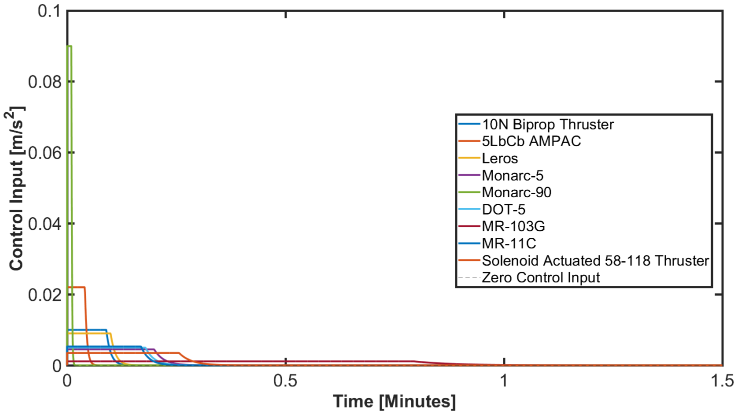

Figure 4-7: Control effort for LQR control gains for bipropellant engines

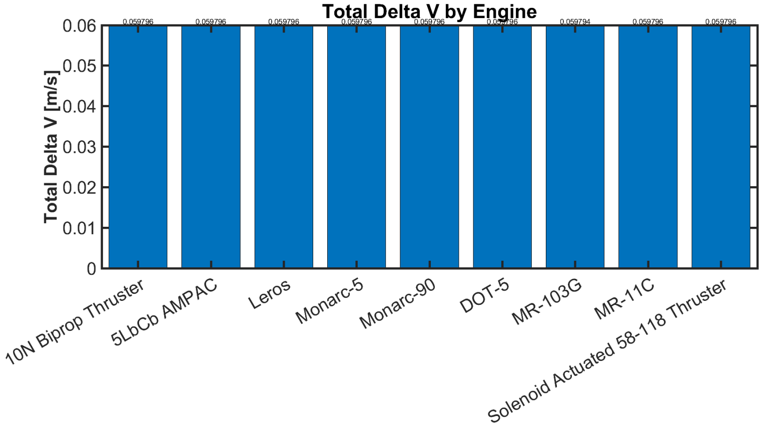

Figure 4-8: ΔV for bipropellant engines

raised altitude state the satellite is commanded to return to the nominal altitude. The satellite is commanded to keep zero along-track delay throughout the maneuver. The altitude and along-track responses are displayed in Figure 4-9 and Figure 4-10 respectively. Figure 4-11 shows the control action implemented by the controller. The satellite tracks the altitude command well and minimizes the along-track error to 0.09 seconds, which is less than 1% of the total slot size. The movement of the maneuver relative to the slot size is shown in Figure 4-12. The maneuver is also plotted relative to time in Figure 4-13.

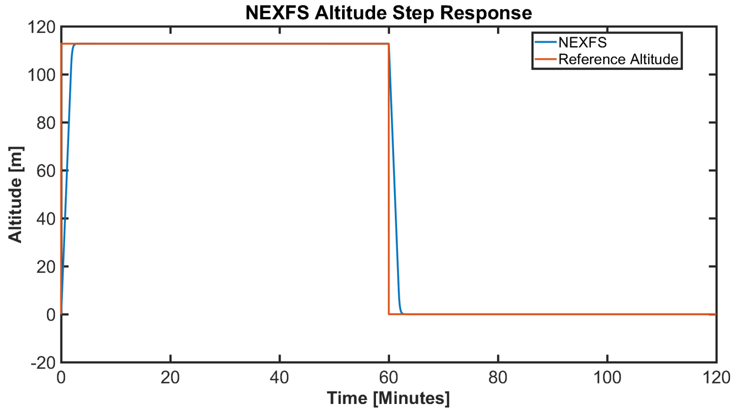

Figure 4-9: NEXFS engine altitude step response with altitude controller

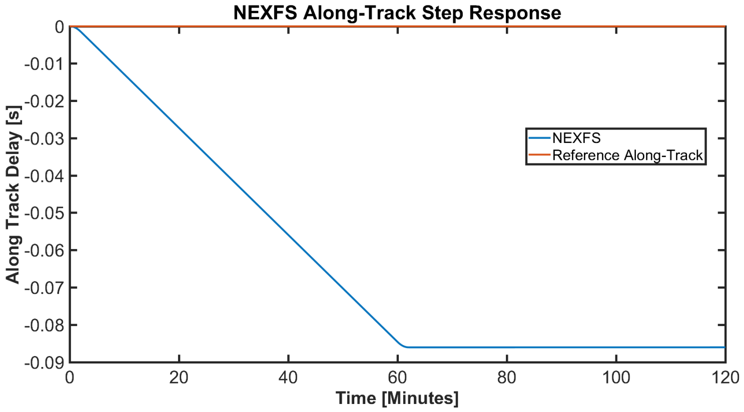

Figure 4-10: NEXFS along-track response with altitude controller

The altitude controller is used to avoid a collision by varying the satellite's altitude in the slot. It is also possible to avoid a collision by moving left or right in the slot by changing the along-track delay. Due to the dynamics shown in Equation 2.10,

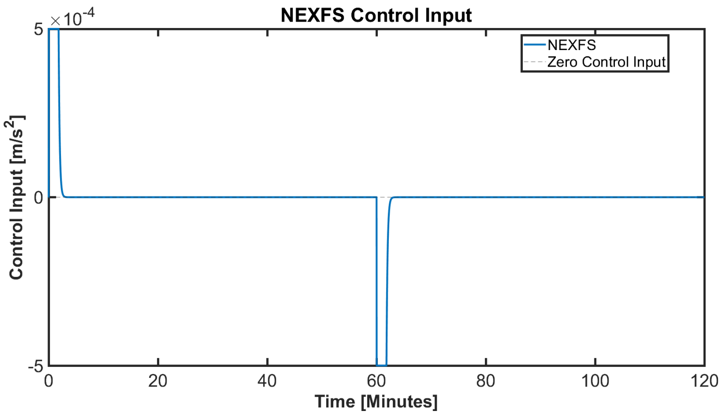

Figure 4-11: NEXFS control input response with altitude controller

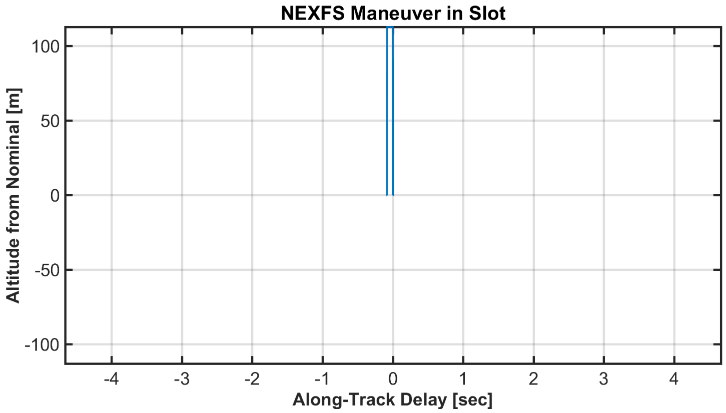

Figure 4-12: NEXFS slot maneuver with altitude controller

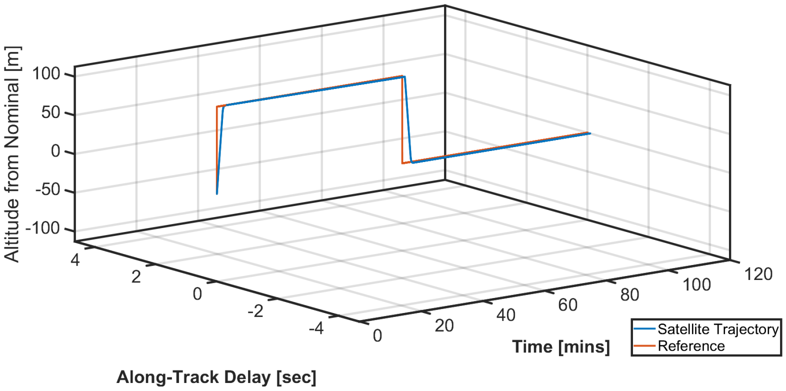

Figure 4-13: NEXFS maneuver with time

the along-track rate of change is related only to the semi-major axis. This means that the along-track delay can only be influenced by the means of adjusting the semi-major axis. This presents a difficulty in control because if a specific along-track delay is needed in the slot, the altitude needs to be planned carefully to achieve that along-track position. The current controller does not have the capability to plan the trajectory to achieve this. Depending on the controller weights, the satellite might leave the slot due to the weights being too high or it might not be able to execute the maneuver due to the weights being too low. Both of these issues were seen while using the altitude controller and applying non-zero along-track references. An example of applying an along-track command to this altitude controller is shown in Figure 4-14. The satellite was commanded to keep the altitude at the nominal value and to move the along-track to the maximum value, x_{2max}. The satellite remains in the slot but only because the satellite fails to follow the command response at all. This is due to the dynamics. Without being able to directly influence the along-track, the controller must adjust the altitude and with the weighting penalizing altitude error it does not react.

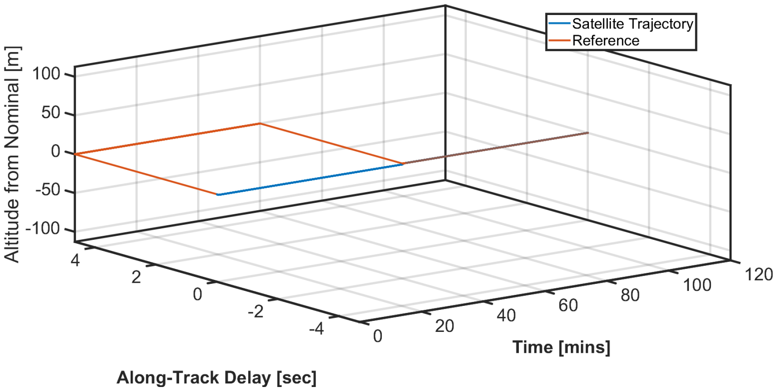

Figure 4-14: NEXFS maneuver using altitude controller and along-track command

4.1.2 Along Track Controller

The along-track controller consists of an LQR controller that is weighted to penalize the along-track state 10 times more than the control input and 100 times more than the altitude state. The control thrusts are capped the same way as the altitude controller. These controllers were only applied to the ion engines as the performance of these controllers was worse than the altitude controllers as applied to the ion engines.

These controllers run at 1 Hz. In general, the along-track controllers overshot the setpoint, took longer to reach the setpoint, and used more ΔV to perform the same maneuver when compared to the altitude controller. The same test maneuvers are done in this section in order to compare the altitude and along-track controllers.

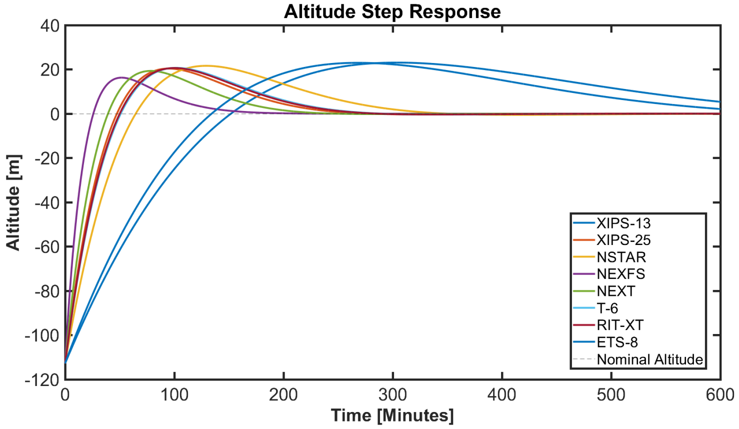

Figure 4-15: Altitude response using an along-track controller with altitude inputs

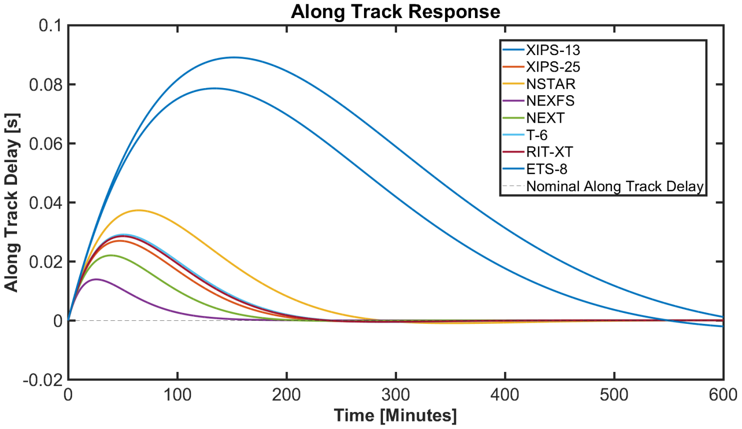

Figure 4-16: Along-track response using an along-track controller with altitude inputs

To make a comparison, we set the satellite at an initial condition at the bottom of the slot with the altitude set to $-x_{1max}$ and the along-track set to 0. The altitude trajectories of the ion engines are depicted in Figure 4-15. NEXFS, the engine with the largest thrust, reaches the setpoint in around 200 minutes, whereas the engines with the least thrust take over 600 minutes. This is because the controllers are now prioritizing the along-track error rather than the altitude error. The along-track error is shown in Figure 4-16. Even though the along-track controller weighs the along-track error more heavily than the altitude controller, the along-track errors for the along-track controller are higher than the altitude controller demonstrated in Figure 4-2. This is due to the dynamics of the system. Since the controller seeks to minimize the along-track error, it neglects the altitude. However, since the altitude directly controls the along-track state, larger errors in the along-track are generated due to the altitude error. This overshoot also results in larger ΔV costs as seen in Figure 4-18. Figure 4-17 shows the control input of the engines. The ΔV costs vary between engines; those with larger overshoot expend more fuel to counteract the overshoot and thus have higher costs. It is immediately seen that the along-track controller is inferior to the altitude controller. It has a longer settling time, overshoots the setpoint, has larger along-track error, and higher ΔV costs.

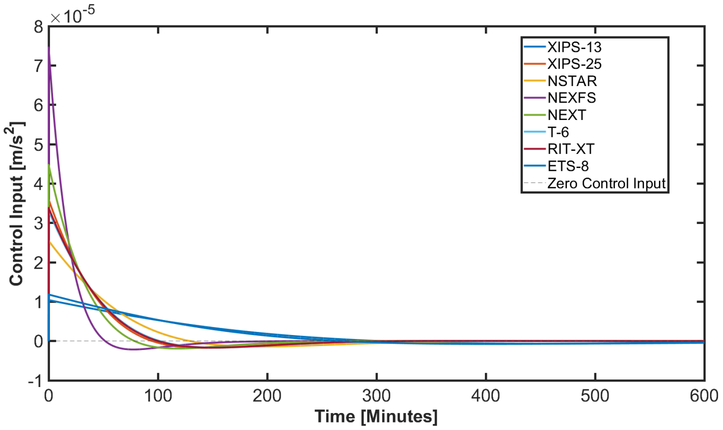

Figure 4-17: Control effort for engines using along-track controller with altitude inputs

Other avenues explored included using along-track inputs to the controller, and using the natural dynamics to save ΔV. Using along-track inputs to the controller rather than altitude inputs resulted in worse performance, and even, in some cases, taking the satellite out of the slot. As mentioned before, the different gains of the controller will cause this behavior in the closed-loop and should be closely examined and tested before implementation. Using the natural dynamics of orbital decay resulted in matching the ΔV costs of the altitude controller but resulted in a longer time to reach the setpoint since the altitude state was not weighted most heavily. In

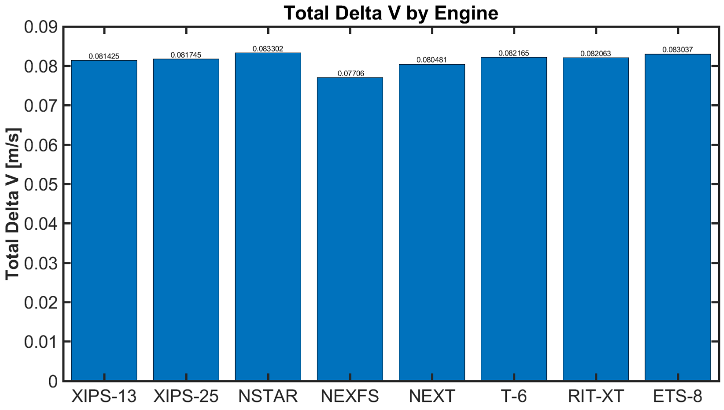

Figure 4-18: ΔV for ion engines using along-track controller

all metrics the altitude controller outperformed the along-track controller so the rest of this study is focused on the altitude controller.

4.1.3 Errors and Disturbances

The simulation of the effects of errors and disturbances is of utmost importance since real systems do not have exact and perfect information. Tests were done to determine how robust these LQR altitude controllers was to errors and disturbances. A simulated +/-5% error of the total slot size (112.75 m) was applied to the sensing of the state. The thruster experienced a +/-10% disturbance of its commanded value while not being able to exceed its maximum thrust capability. Errors and disturbances were generated using a uniform distribution. The controller read the state input with error and generated a thrust command. The thrust command was sent to the engine and then experienced the disturbance.

Tests were conducted using this method of generating errors and disturbances. A satellite was left at the nominal orbit altitude and along-track and the thrusters were set to maintain this nominal position. Note that the controller is linearized at the nominal orbit altitude so all of the ΔV in use here is to combat errors and disturbances. There is a very small error in the altitude and along-track states, as can be seen in Figures 4-19 and 4-20. The dashed lines in these figures represent the slot boundaries. Note that larger errors are seen in engines with higher thrust since

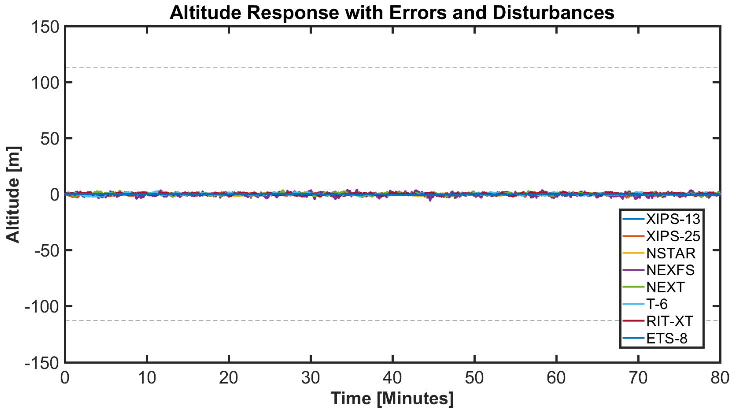

Figure 4-19: Satellite altitude with errors and disturbances

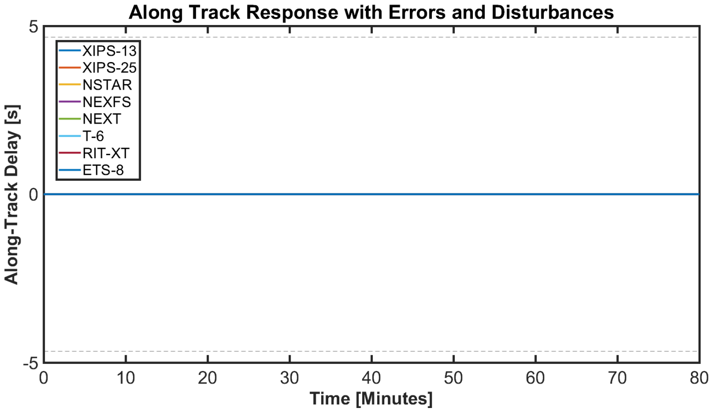

Figure 4-20: Satellite along-track with errors and disturbances

disturbances are applied as a percentage of the thrusters' commanded input.

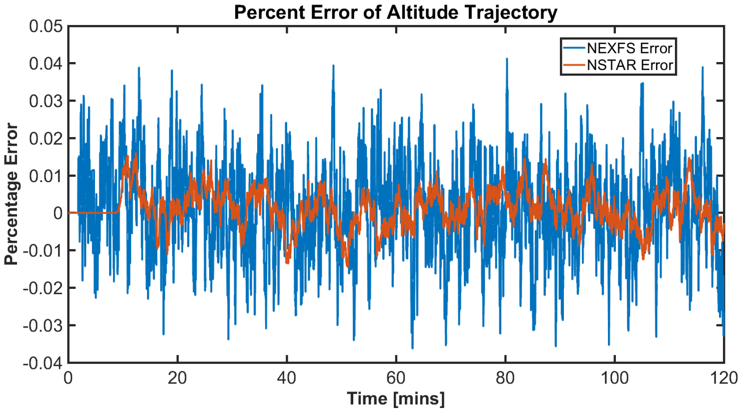

Figure 4-21: Percentage error of satellite altitude

The disturbances and errors are also applied to a maneuver where the satellite is

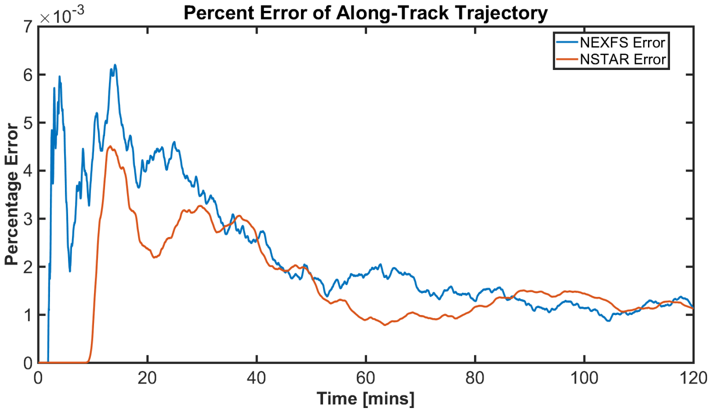

Figure 4-22: Percentage error of satellite along-track

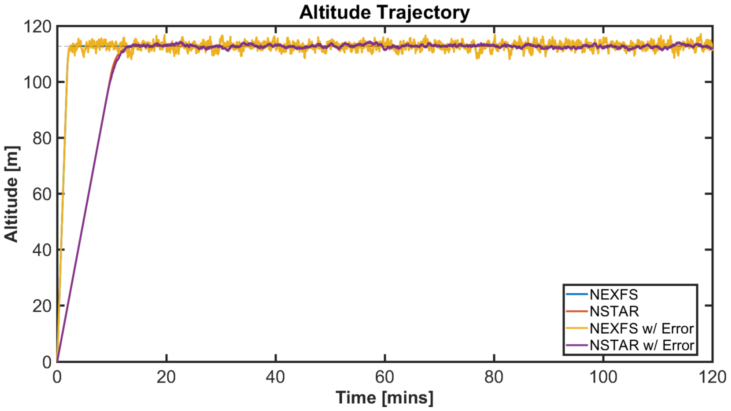

Figure 4-23: Altitude maneuver with error

commanded to move from the nominal position to the top of the slot. The maneuver is applied to two ion engines, NSTAR, which has a maximum 92 mN of thrust, and NEXFS, which has a maximum of 500 mN of thrust. The solutions without error and the solutions with error are compared by creating a percentage error. The altitude percentage error is less than 0.05% for both the NEXFS and NSTAR engines. The along-track errors for both engines measure less than 0.007%. The figures for the altitude and along-track percent errors are shown in Figure 4-21 and Figure 4-22. Despite the controller being able to reject these errors, the controller does not guarantee that response of the satellite will stay within the slot. For instance, Figure 4-23 shows the control of a satellite that was commanded to the edge of the slot. Despite reaching this setpoint, the errors and disturbances push the satellite out of the slot at times. The dashed line represents the edge of the altitude slot. It is possible that these types of controllers keep the satellite within the slot but it is not guaranteed. Due to this issue, we turn our attention to the tube MPC method.

4.2 Tube MPC Methods

The tube MPC controller consists of a disturbance-rejection controller and an MPC to ensure the satellite stays within the slot [23]. The controller uses LQR gains that were found via Bryson's rule and used the same weights as the altitude controller in Table 3.3. The controllers were applied to both ion and bipropellant engines. First, the controller as applied to a specific ion engine is examined and then to a specific bipropellant engine.

The NEXFS ion engine is selected for having the highest thrust in its class of ion engines examined here. The controller is running at a 1 Hz with a prediction horizon of 10 steps. The uncertainty in the state for this simulation is 1% of the total slot size. This uncertainty is applied as a uniform distribution. The thruster experiences disturbances in the control input equal to +/-10% of the requested control action. All of these uncertainties and disturbances are applied using a standard distribution.

The first simulation is meant to simulate an avoidance maneuver for the satellite. It begins the simulation at the middle of the slot and is commanded to reach the maximum altitude state while maintaining zero along-track delay. The simulation is run for 150 seconds. The tube plots are omitted from the figure to keep the figure clean. The simulation can be found in Figure 4-24. The controller raises the altitude of the satellite but is unable to reach the setpoint. This is because the tube MPC ensures that the satellite cannot leave the slot. We see that the altitude settles at around 100 meters, which is 12.75 m less than the commanded altitude of the maximum state. The satellite's movement at the end of the maneuver is due to the uncertainty in the state. The controller is still calculating control action at this point but the thrusts values are an order of magnitude smaller than when the controller was actively trying to change the altitude. At this point in the maneuver the thrust is being used solely to manage the uncertainty and disturbances.

The next simulation is to quantify the amount of error in the controller maintaining the satellite's position in the middle of the slot. The simulation parameters are the same except that the simulation is only run for 100 seconds and the satellite is commanded to stay at the center of the slot. The percentage error was calculated by dividing the altitude by x_{1max} and dividing the along-track by x_{2max} and multiplying by 100. The simulation is shown in Figure 4-25. The initial tube is shown in the figure and the setpoint of [0,0] is omitted to maintain visibility. The controller acts against the disturbances and uncertainty to maintain the satellite at the setpoint. Figure 4-26 and Figure 4-27 show the percent error of the maneuver. Both the altitude and along-track maintain less than a 5% error throughout the simulation.

It is also important to quantify if the tube MPC controller has the ability to follow along-track commands. Due to the dynamics described in Equation 2.10 the along-track delay can only be influenced by the altitude. In order for the controller to follow along-track commands, it must adjust the altitude state. To ensure that the controller has enough time to complete these commands, the theoretical time it takes to complete the maneuver is calculated. It is known from Equation 2.10 that:

$$\dot{\Delta T} = \frac{-3}{2a_0}a \tag{4.1}$$

The maximum rate for $\dot{\Delta T}$ is reached when the altitude is set to x_{1max}. Therefore, the maximum rate can be described by:

$$\Delta\dot{T}_{max} = \frac{-3}{2a_0}x_{1max} \tag{4.2}$$

Now to find the time it takes to reach the maximum along-track state x_{2max}, we divide the maximum along-track state by the maximum rate and take the absolute value.

$$t_{req} = |\frac{x_{2max}}{\Delta\dot{T}_{max}}| \tag{4.3}$$

This results in t_{req} being equal to $1.9512 * 10^5$ seconds. The simulation is set to $4.0 * 10^5$ seconds to allow the controller enough time to act. This is more than double the required amount of time the controller would need to raise the orbit, wait for the along-track to change, and then lower the orbit altitude to reach the setpoint. The frequency of the controller is decreased to 0.005 Hz reduce computation time. The prediction horizon is also increased to N = 200 to ensure the controller can "see" sufficiently far enough ahead. The controller is commanded to reach the setpoint of $[0, -x_{2max}]$. The controller should increase the altitude, wait for the natural dynamics to change the along-track, and then decrease the altitude back to the central position. The simulation is shown in Figure 4-28. The controller is not able to foresee the movements it needs to take to reach the desired along-track position. The satellite state drifts over time due to the accumulation of state uncertainty.

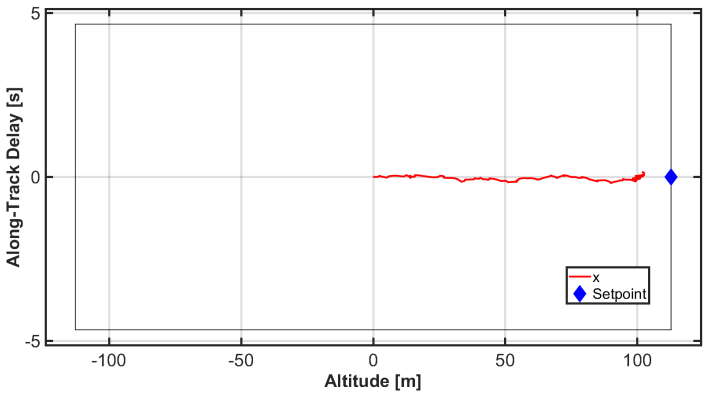

Figure 4-24: NEXFS tube MPC altitude command response

Now similar tests are run but for the Monarc-90 bipropellant engine which has 90 Newtons of thrust. The controller is running at 1 Hz with a prediction horizon of 10 steps. The thruster experiences disturbances in the control input of +/-10% of the requested control action. Two simulations were run with different uncertainty percentages in the state. Figure 4-29 shows the controller's response with a state uncertainty of 2.5% of the total slot size. Figure 4-30 shows the controller's response with a state uncertainty of only 1%. Comparing these two figures shows the influence of the state uncertainty on the tube. Figure 4-30 shows how much closer the satellite can get to the commanded input when there is less uncertainty. The satellite's altitude is about 100m, as compared to Figure 4-29 where the satellite is only about to reach about 80m.

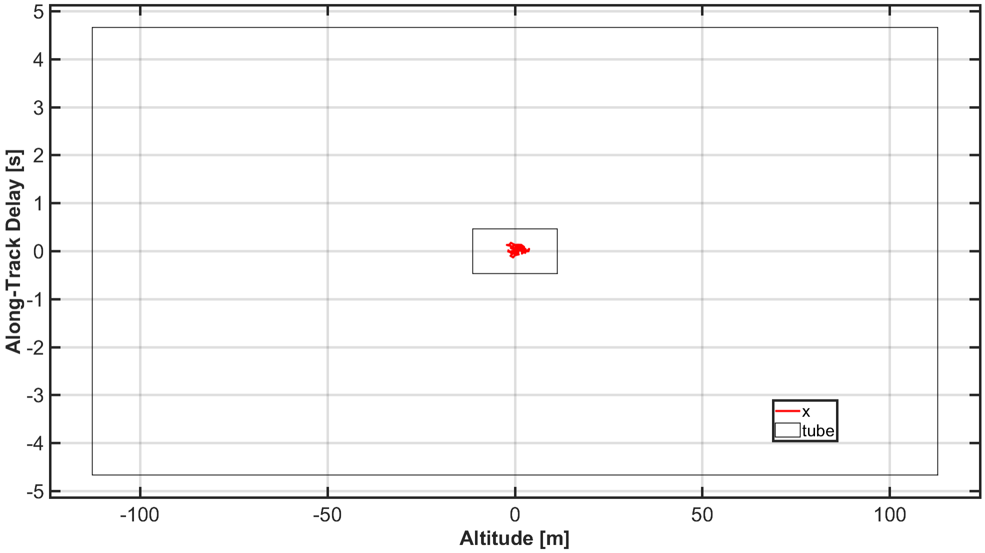

Figure 4-25: NEXFS tube MPC maintaing position in center of slot

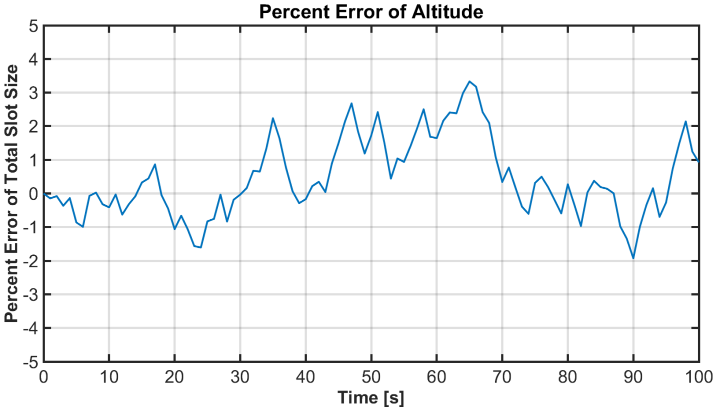

Figure 4-26: NEXFS tube MPC altitude error

Simulations are also run to quantify the percentage error for the Monarc-90 engine. Once again, the simulation parameters are the same except that this simulation is only run for 100 seconds and the satellite is commanded to stay at the center of the slot. The state uncertainty error is left at 1% to be able to compare results to the ion engine previously examined. The percentage error was calculated in the same way, simply by dividing the altitude by x_{1max} and the along-track by x_{2max} and multiplying by 100. The maintenance maneuver is shown in Figure 4-31 and the errors in altitude and along-track are shown in Figure 4-32 and Figure 4-33 respectively. The controller

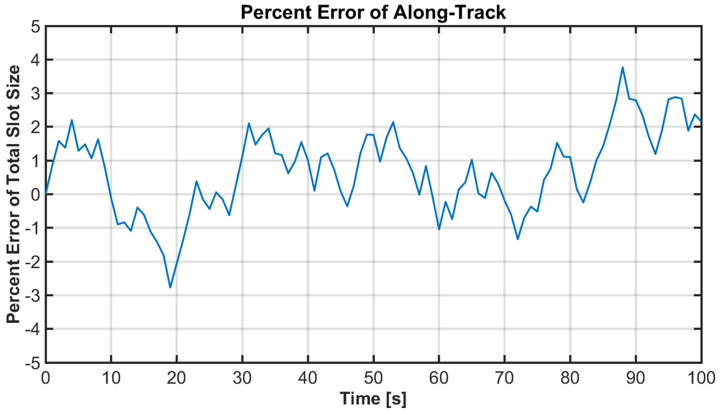

Figure 4-27: NEXFS tube MPC along-track error

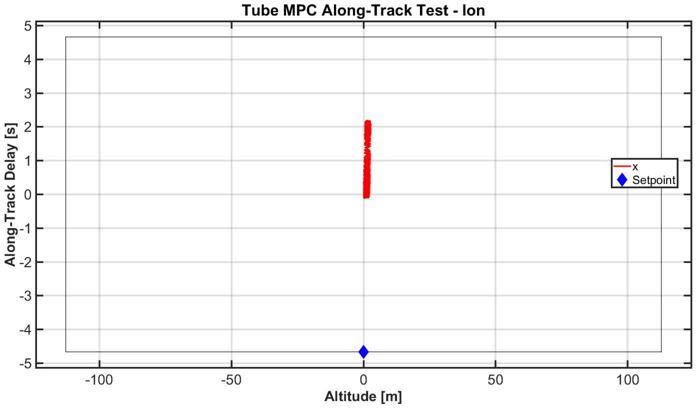

Figure 4-28: NEXFS tube MPC along-track test

is able to maintain the altitude error under 1% throughout the simulation. The along-track error grows to above 5% but this is due to uncertainty in the position rather than the controller's response.

Another simulation is done to determine if the bipropellant engine controller used for the Monarc-90 engine has the capability to complete along-track commands without planning. The simulation is once again set to $4.0 * 10^5$ seconds, the frequency of the controller is set to 0.005 Hz, and the prediction horizon is also increased to N = 200. The controller is commanded to reach the setpoint of $[0, -x_{2max}]$. This simulation is shown in Figure 4-34. Once again the controller is not able to foresee the control action it needs to take to reach the desired along-track position. The movement in the figure is not due to any control action, but rather drift due to the uncertainty of the satellite's state.

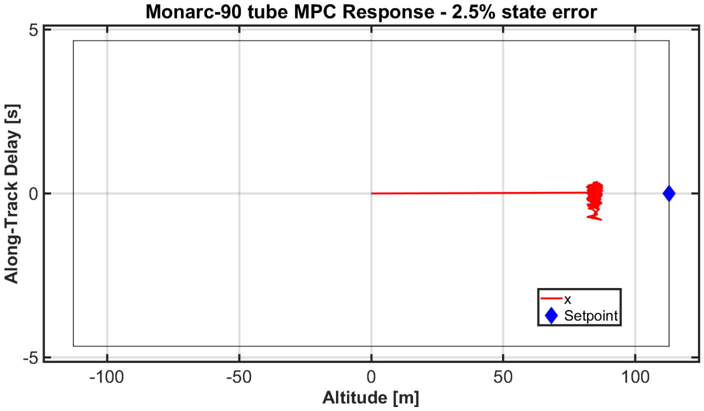

Figure 4-29: Monarc-90 tube MPC altitude command response - 2.5% state error

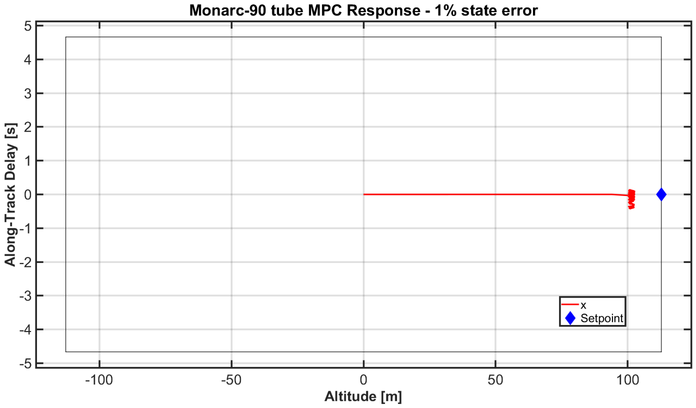

Figure 4-30: Monarc-90 tube MPC altitude command response - 1% state error

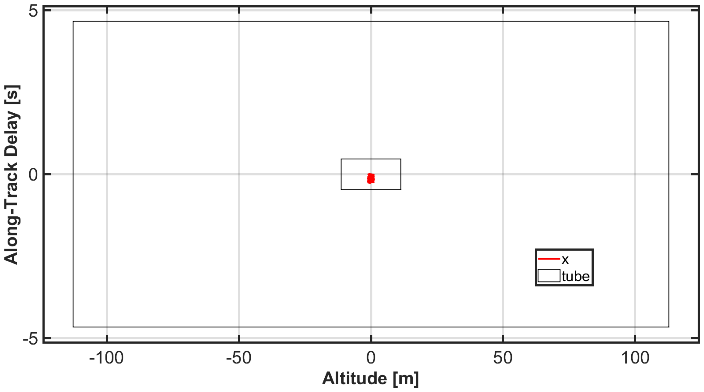

Figure 4-31: Monarc-90 tube MPC maintaing position in center of slot

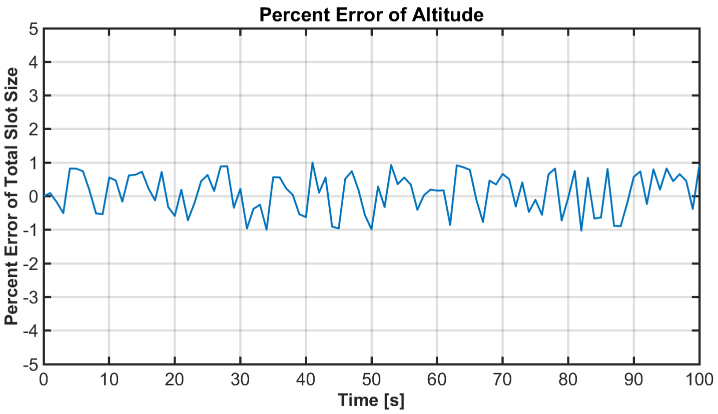

Figure 4-32: Monarc-90 tube MPC altitude error

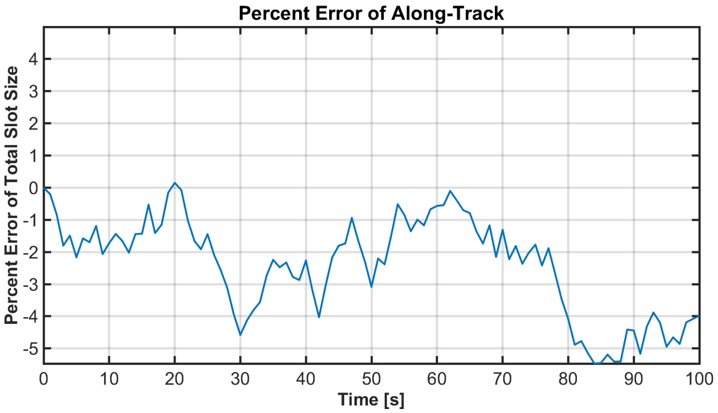

Figure 4-33: Monarc-90 tube MPC along-track error

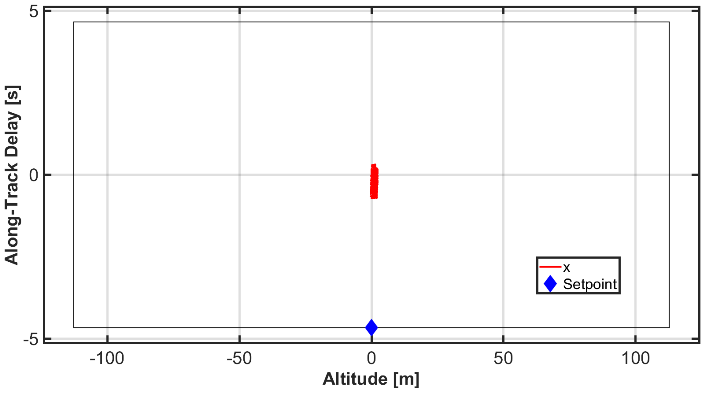

Figure 4-34: Monarc-90 tube MPC along-track test

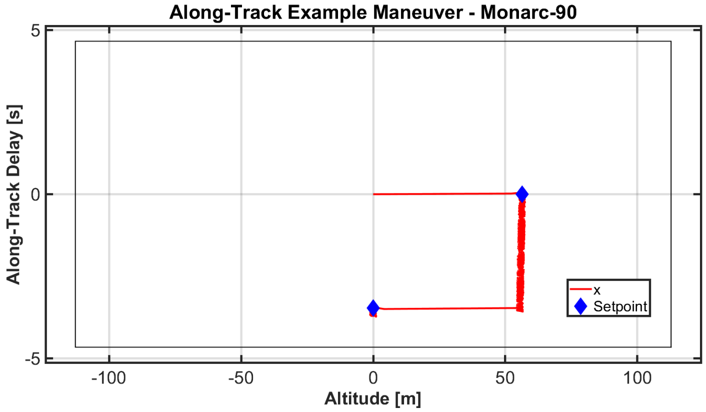

Figure 4-35: Along-track example maneuver

4.3 Discussion

This research focused on testing both LQR controllers with different weighting schemes and tube MPC controllers to the problem of maintaining satellite orbits and collision avoidance while in satellite constellations frameworks. To that end, five different LQR based controller configurations were tested. 1. An altitude-weighted controller that took altitude commands. 2. An altitude-weighted controller that took along-track commands. 3. An along-track weighted controller that took altitude commands. 4. An along-track weighted controller that took along-track commands. 5. An along-track weighted controller that utilized the natural dynamics of the system to save on ΔV that took altitude commands. These tests will be described but the results ultimately informed of a need to develop tube MPC controllers to ensure that the satellite is mathematically guaranteed to stay within the slot, even when it is commanded to go to the edge of the slot. The tube MPC controllers accomplished this goal but ultimately lack the ability to track along-track commands without prior trajectory planning. The assurance of staying within the slot also comes at the cost of other metrics that will be described later in this section.

The LQR altitude controller exhibited the fastest and most accurate altitude tracking, as a result had the lowest along-track error, and consumed the least fuel. The altitude controller responded best when taking only altitude inputs and does not overshoot the setpoint. The altitude controller using along-track inputs did not leave the slot in testing but theoretically could leave the slot depending on the specific controller weights. In this testing, the controller simply did not respond due to its inability to utilize the inherent dynamics of the problem to adjust the along-track state through manipulating the altitude state. The controller is also not guaranteed to keep satellites within the slot.

The along-track controller had multiple issues. It had poor response time, overshoot issues, large along-track errors when compared to the altitude controller, and larger fuel costs. It also left the slot when commanded to the edge of the slot. When this controller was combined with along-track inputs, it responded with even worse performance and left the slot even further than when the controller took altitude inputs. When the controller used the natural orbital decay to minimize ΔV, the controller had a poor response time when compared to the altitude controller but did not overshoot and had the same fuel costs as the altitude controller. The along-track error increased due to the controller's poor response time. In addition, since the controller is prone to overshooting and there is no guarantee of satellite placement within the slot, it is not safe to use for satellite constellations as it has the possibility to cause conjunctions.

The altitude controller was robust to errors and disturbances. A continuous uniform distribution was used to create errors in the sensed state and output thrust. A +/-5% state error and a +/-10% thruster disturbance was reduced to a less than 0.05% error in the altitude and less than 0.007% error in the along-track for the thrusters tested. The along-track errors are a function of the altitude errors in this particular simulation as a satellite keeping its position in the middle of the slot will result in a zero along-track state due to the dynamics. Despite this error rejection, the controller does not provide a guarantee that the satellite will not exit the slot despite these errors.

The altitude controller emerged as the best controller for the purpose of avoiding collisions and having accurate movement within the slot. However, there could be some modifications to the controller to improve it. First, if a collision is avoided and the satellite does not need to return to its nominal setpoint immediately, it should take advantage of the natural dynamics to minimize fuel costs. Another issue with the altitude controller is that it does not allow robust control of the along-track state. Although all states in the slot are reachable as shown by the controllability and observability matrices in Section 2.2.1, the along-track state can only be influenced through the altitude state and this presents a difficulty in control. Along-track commands are desirable because they expand the types of maneuvers that are possible within the slot. If an object is passing through the slot vertically the controller will not be able to dodge the movement without a human planning the trajectory. The planner could raise or lower the orbit of the satellite and let the along-track delay increase or decrease to avoid the collision before the object comes into the slot. This requires planning and knowledge of the potential collision in advance.

In order to improve upon these problems of the LQR altitude controller, multiple tube MPC controllers were developed and applied to various engines. The NEXFS engine was selected as the main ion engine and the Monarc-90 was selected as the main bipropellant engine to run simulations with. This is because these engines have the highest thrust capacities in their class and generally will be better at rejecting disturbances and reaching the setpoints. In addition, the same engines were used to classify the error rejection of the LQR controller so the same engines must be used to compare. The most important result from the tube MPC controller is that the controller successfully guarantees that the satellite is constrained within the slot. This is shown in Figure 4-24, Figure 4-29, and Figure 4-30. Despite the controller being commanded to reach the maximum altitude position, the controller stops short to ensure that the satellite does not leave the slot. The uncertainty in the controller's position influences the stopping point for the satellite. When Figure 4-29 is compared to Figure 4-30, it is easy to see the difference. Although the tube MPC can ensure the satellite does not leave the slot at a state error of 2.5%, the range of reachable states is decreased as a payment for this assurance. Compare this to when the controller assumes the state error is 1%. The controller is able to maintain a closer position to the setpoint because there is less error in the states. Many different errors were used while producing this work, some even up to 5%. However, only these were included in this thesis as errors as large as 5% resulted in a reachable workspace that was very small, sometimes less than half the maximum state values. This can also be adjusted by adjusting the gains of both the disturbance rejection controller (the tube) and the nominal MPC. In this work, the gains were implemented using Bryson's Rule weights from Table 3.3. Different weights will certainly have an effect on the effective reachable states within the tube MPC framework. As a result of this, specific mission objectives and which states the satellite needs to access will be the main factors that influence how accurate and precise the state measurements must be, as well as the control devices chosen for the satellite and the controller gains.

The tube MPC controllers were also tested by commanding them to maintain their position in the center of the slot and quantifying the error in both altitude and along-track. The results for the NEXFS controller's simulation can be found in Figure 4-25 and the errors for the maneuver are shown in Figure 4-26 and Figure 4-27. The controller is able to maintain the satellite's position around the center and maintains the altitude error below 5%. The along-track error is less important in these simulations as the along-track is indirectly controlled by the altitude state. The along-track error is almost purely a result of the error in the state estimation as the satellite is trying to maintain its position in the center of the slot, which would result in no along-track delay under the dynamics considered here. The results for the Monarc-90 simulation are found in Figure 4-31 and the errors are shown in Figure 4-32 and Figure 4-33. The controller is able to keep the satellite in the center of the slot and maintains the satellite's altitude error below 1%. This improved performance as compared to the ion engine is most likely due to the higher thrust of the engine, meaning that the controller is able to better compensate for errors. The error of the along-track exceeds 5% at times but it is important to note that the errors in along-track delay are a result of errors in state estimation. In fact, because the estimation errors are generated using a standard distribution, some simulations would show less percentage error in the along-track. Despite that, the controller was able to maintain the altitude error around the area of 1%.

It was also of interest to determine if the tube MPC was capable of following along-track commands. The results of these tests are depicted in Figure 4-28 for the ion engine and in Figure 4-34 for the bipropellant engine. For these simulations, the simulation time was increased to $4.0 * 10^5$ seconds to give the controller enough time to respond. The frequency was decreased to 0.005 Hz and the prediction horizon was increased to N = 200. This means that the controller is able to see $4 * 10^4$ seconds into the future, which is 10% of the total simulation time. The satellite was commanded to reach the setpoint of $[0, -x_{2max}]$. Both controllers failed to respond to the along-track command. An ideal maneuver would result in the satellite raising its altitude, waiting for the dynamics to adjust the along-track, and then lower the altitude to reach the desired input. Depending on the maneuver, the satellite could also lower its altitude, wait, and then raise the altitude to reach the desired along-track state. This deficiency points to a need for a trajectory generator. Since the along-track dynamics are only influenced by the altitude, planning out the altitude trajectory with a trajectory generator would give the controller the ability to reach the states required by along-track commands. An example of this type of maneuver is provided in Figure 4-35. Instead of letting the controller plan the maneuver, the controller was fed two different setpoints with higher level planning. First, the controller was commanded to reach the first-setpoint at $[\frac{x_{1max}}{2},0]$ and wait while the along-track state changed. Once the along-track state reached about $\frac{-3x_{2max}}{4}$, the controller was commanded to decrease the altitude. The controller was able to reach the overall setpoint of $[0,\frac{-3x_{2max}}{4}]$ but it required higher level planning. A trajectory generator would give the controller the ability to reach along-track states by segmenting the setpoints in this fashion.

There are many improvements that could be made to make this controller framework more robust and prepared for real operation. For future work, a trajectory generator should be implemented to allow the controller to fully access the along-track states. The controller is also measuring the state each time it calculates a control action. In real life usage, the controller might not be able to receive state measurements at high frequencies so an estimator could be applied to deal with incomplete information. Not only that, the estimator could also improve the errors within the sensors as repeated measurements and internal calculations by the satellite could drive the estimated state closer to reality. A model of applying the controller to an entire satellite constellation could demonstrate the use of this controller in an actual constellation framework, such as a slotting architecture to optimize orbital volume [7]. Satellites could be commanded to move around within the slots to demonstrate this. Having knowledge of potential collisions in advance will allow engineers to opt for low-thrust engines in their designs, potentially reducing weight, cost, mass, and volume. However, the quick reaction time proven by some of the simulations also points to the possibility of satellites being equipped with low-range sensors that sense the proximity of other objects within the satellites' area. These sensors might be able to see debris or objects too small to be observed from Earth and could enable the satellite to avoid collisions that would otherwise be unavoidable.

Chapter 5

Conclusion

The work in this thesis describes a linearized, simplified model for satellite altitude control and applies two known controller methods to that end. Both an LQR controller and a tube MPC controller are applied to the problem of satellite altitude control and collision avoidance within a satellite constellation framework and evaluated. The controllers are shown to control the altitude of the satellite well but lack the ability to carry out along-track commands without a proper trajectory. The tube MPC controller is able to guarantee the placement of the satellite within the slot. It is also shown that is is possible to linearize the dynamics model by performing a Taylor series expansion around the operating point while maintaining extraordinary accuracy in the model.

The performance of the LQR controller is assessed for different study cases. Multiple LQR controllers for multiple engines are generated using different weights on the along-track and the altitude states. It was found that the altitude weighted controller resulted in the most accurate altitude tracking, least along-track error, and had the lowest ΔV costs. In addition, the altitude controller is robust to input and output errors. The introduction of a +/-5% error on the state and a +/-10% disturbance on the thrust resulted in a less than 0.05% error on the altitude state and 0.007% error on the along-track state for the thrusters and maneuvers tested. Despite these benefits, none of the LQR controllers could guarantee placement within the slot and depended on the user to intelligently plan maneuvers. In addition, the LQR controllers could not follow along-track commands.

This resulted in the need to develop a tube MPC controller for the problem of satellite collision avoidance and orbit maintenance within a slot satellite constellation framework. The tube MPC guaranteed placement within the slot. Decreasing the uncertainty in the state estimation allowed the satellite to access more of the slot. Increasing the uncertainty in the state estimation lowered the satellite's access to the slot. Two engines were examined to quantify the error. The NEXFS ion engine controller resulted in a percent error of around 3% while maintaining its position whereas the Monarc-90 engine controller, which has more thrust, resulted in a 1% error. It is important to understand these results are dependent on the tuning of the controller and on the random behavior of the error implemented in the state estimation. The tube MPC controller was not able to follow along-track commands unless they were segmenting, which points to a need for a trajectory generator.

Future work will include the development of a trajectory generator so that the controller is able to follow along-track references. Developing an estimator to go along with the controller will ensure that the satellite does not need constant state measurements as it can be estimated. The estimator will also allow the satellite to lower the error of the state if it has the ability to make repeated measurements. The fast response of these controllers also point to the satellite's possible ability to avoid collisions that are caused by small debris or objects unable to be observed from Earth if equipped with sensors that can sense objects within a close proximity.

Bibliography

[1] D. Arnas, P. Jurado, I. Barat, B. Duesmann, and R. Bock. Flex: A parametric study of its tandem formation with sentinel-3. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2447–2452, 2019.

[2] David Arnas. Linearized model for satellite station-keeping and tandem formations under the effects of atmospheric drag. *Acta Astronautica*, 178:835–845, 2021.

[3] David Arnas and Daniel Casanova. Nominal definition of satellite constellations under the earth gravitational potential. *Celestial Mechanics and Dynamical Astronomy*, 132:1–20, 2020.

[4] David Arnas, Daniel Casanova, and Eva Tresaco. Relative and absolute station-keeping for two-dimensional–lattice flower constellations. *Journal of Guidance, Control, and Dynamics*, pages 2602–2604, 2016.

[5] David Arnas, Daniel Casanova, and Eva Tresaco. 2D Necklace Flower Constellations. *Acta Astronautica*, 142:18–28, 10 2017.

[6] David Arnas, Daniel Casanova, and Eva Tresaco. Time distributions in satellite constellation design. *Celestial Mechanics and Dynamical Astronomy*, 128(2-3):197–219, 2017.

[7] David Arnas, Miles Lifson, Richard Linares, and Martin Avendaño. Definition of low earth orbit slotting architectures using 2d lattice flower constellations. *Advances in Space Research*, 2020.

[8] Martín E. Avendaño, Jeremy J. Davis, and Daniele Mortari. The 2-D lattice theory of Flower Constellations. *Celestial Mechanics and Dynamical Astronomy*, 116(4):325–337, 8 2013.

[9] I. Barat, F.J. Sanchez, B. Duesmann, D. Arnas, P. Jurado, and R. Bock. Flex tandem with sentinel-3: Control concept. *18th Australian International Aerospace Congress*, 18, 2018.

[10] Claudio Bombardelli and Javier Hernando-Ayuso. Optimal impulsive collision avoidance in low earth orbit. *Journal of Guidance, Control, and Dynamics*, 38(2):217–225, 2015.

[11] Mario Melendrez Contreras, David Arnas, and Richard Linares. State-space controller for leo, low-thrust satellite constellations. *American Astronautical Society*, 2021.

[12] John E. Draim. A common-period four-satellite continuous global coverage constellation. *Journal of Guidance, Control, and Dynamics*, 10(5):492–499, 1987.

[13] Caleb Henry. *Amazon planning 3,236 satellite constellation for internet connectivity*, 2019. (last accessed on May 12, 2021) `https://spacenews.com/amazon-planning-3236-satellite-constellation-for-internet-connectivity`.

[14] Donald J. Kessler, Nicholas L. Johnson, J.-C. Liou, and Mark Matney. The kessler syndrome: Implications to future space operations. *American Astronautical Society*, 137, 2010.

[15] Gerhard Krieger, Alberto Moreira, Hauke Fiedler, Irena Hajnsek, Marian Werner, Marwan Younis, and Manfred Zink. Tandem-x: A satellite formation for high-resolution sar interferometry. *IEEE Transactions on Geoscience and Remote Sensing*, 45(11):3317–3341, 2007.

[16] Huibert Kwakernaak and Raphael Sivan. *Linear optimal control systems*, volume 1. Wiley-interscience New York, 1972.

[17] S Le May, S Gehly, BA Carter, and S Flegel. Space debris collision probability analysis for proposed global broadband constellations. *Acta Astronautica*, 151:445–455, 2018.

[18] C.L. Leonard, W.M. Hollister, and E.V. Bergmann. Orbital formationkeeping with differential drag. *Journal of Guidance Control and Dynamics*, 12(1):108–113, 1989.

[19] Stephen Leroy, Riley Fitzgerald, Kerri Cahoy, James Abel, and James Clark. Orbital maintenance of a constellation of cubesats for internal gravity wave tomography. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:307–317, 2020.

[20] J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[21] Romain Lucken and Damien Giolito. Collision risk prediction for constellation design. *Acta Astronautica*, 161:492–501, 2019.

[22] R. David Luders. Satellite Networks for Continuous Zonal Coverage. *ARS Journal*, 31(2):179–184, 2 1961.

[23] D.Q. Mayne, M.M. Seron, and S.V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[24] David Mishne and Eviatar Edlerman. Collision-avoidance maneuver of satellites using drag and solar radiation pressure. *Journal of Guidance, Control, and Dynamics*, 40(5):1191–1205, 2017.

[25] J Moreno, Y Goulas, A Huth, E Middleton, F Miglietta, G Mohammed, L Nedbal, U Rascher, W Verhoef, M Drusch, et al. Report for mission selection: Flex. *ESA SP*, 1330(2), 2015.

[26] D. Mortari, M.P. Wilkins, and C. Bruccoleri. The Flower Constellations. *Journal of the Astronautical Sciences*, 52:107–127, 2004.

[27] Russell P Patera. General method for calculating satellite collision probability. *Journal of Guidance, Control, and Dynamics*, 24(4):716–722, 2001.

[28] Russell P Patera. Satellite collision probability for nonlinear relative motion. *Journal of Guidance, Control, and Dynamics*, 26(5):728–733, 2003.

[29] Jonas Radtke, Christopher Kebschull, and Enrico Stoll. Interactions of the space debris environment with mega constellations—using the example of the oneweb constellation. *Acta Astronautica*, 131:55–68, 2017.

[30] S.V. Rakovic, E.C. Kerrigan, K.I. Kouramas, and D.Q. Mayne. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50(3):406–410, 2005.

[31] L Rider. Optimized polar orbit constellations for redundant earth coverage. *Journal of the Astronautical Sciences*, 33:147–161, 1985.

[32] M. Rubenstein and Z. Manchester. Bio-inspired position control of satellite constellations. *Distributed Autonomous Robotic Systems. Proceedings in Advanced Robotics*, 9:441–450, 2019.

[33] Adrian Schubert, Nuno Miranda, Dirk Geudtner, and David Small. Sentinel-1a/b combined product geolocation accuracy. *Remote sensing*, 9(6):607, 2017.

[34] GL Slater, SM Byram, and TW Williams. Collision avoidance for satellites in formation flight. *Journal of guidance, control, and dynamics*, 29(5):1140–1146, 2006.

[35] SpaceX. *Starlink*, 2019. (last accessed on May 14, 2020) `https://www.starlink.com`.

[36] G.L. Stephens, D.G. Vane, R.J. Boain, G.G. Mace, K. Sassen, Z. Wang, A.J. Illingworth, E.J. O'Connor, W.B. Rossow, S.L. Durden, S.D. Miller, R.T. Austin, A. Benedetti, C. Mitrescu, and the Cloudsat Science Team. The cloudsat mission and the a-train: A new dimension of space-based observations of clouds and precipitation. *Bulletin of the American Meteorological Society*, 83(12):1771–1790, 2002.

[37] Yuri Ulybyshev. Long-term formation keeping of satellite constellation using linear-quadratic controller. *Journal of Guidance, Control, and Dynamics*, 21(1):109–115, 1998.

[38] J.G. Walker. Satellite Constellations. *J. Br. Interplanet. Soc.*, 37:559–572, 1984.

[39] David F. Everett Wertz, James Richard and Jeffery John Puschell. *Space mission engineering: the new SMAD.* Microcosm Press, 2011.

[40] J.R. Wertz, J.T. Collins, S. Dawson, H.J. Koenigsmann, and C.W. Potterveld. *Autonomous constellation maintenance.*, pages 263–273. Springer, Dordrecht, 1998.