Electric Propulsion Intelligent Control (EPIC) Toolbox for Proximity Operations and Safety Analysis in Low-Earth Orbit (LEO)

Axel Garcia
Massachusetts Institute of Technology
Richard Linares
Massachusetts Institute of Technology

Zachary Folcik
MIT Lincoln Laboratory

Peter Niedfeldt
MIT Lincoln Laboratory

James Gandek
MIT Lincoln Laboratory

Christopher Jewison
The Charles Stark Draper Laboratory, Inc

Paul Cefola
University at Buffalo (SUNY)

ABSTRACT

The main goal of this research is to build an optimal toolset to enable mission trajectory planning using low-thrust platforms. More specifically, the Electric Propulsion Intelligent Control (EPIC) project aims to analyze operations including low-thrust transfers, chaser satellite orbiting with Natural Motion Circumnavigation (NMC) around a target, and rendezvous and flybys aimed at proximity operations in LEO. In this paper, we show analytical expressions for solving low-thrust proximity operations between satellites and results of the various optimization phases via the GPOPS-II optimization package. Additionally, we provide controller-agnostic safety analysis based on control barrier functions to certify collision avoidance during proximity operations, even under unknown but bounded perturbations or non-cooperative disturbances. This project has numerous guidance and control applications for achieving mission success and safety in autonomous operations.

1. INTRODUCTION

The inspection and observation of low-Earth-orbit resident space objects (RSOs) is becoming increasingly important for a wide range of applications [17]. These space operations can enable more sustainable capabilities for debris retrieval, on-orbit assembly and servicing needed due to the rapid increase of RSOs at these altitudes. Furthermore, maneuvers to place a chaser spacecraft into an elliptical path around a target, such as Natural Motion Circumnavigation (NMCs), are often used. This maneuver technique allows the inspection of a target satellite from a variety of aspect angles, and it is considered as a passively-safe inspection technique [12]. For a chaser satellite characterizing a target satellite in Low-Earth Orbit (LEO), time-optimal and fuel-optimal maneuvers are key to improving the capability and mission life of the target satellite. Continuous low-thrust burns are presented in this paper, as opposed to calculating burns as near impulsive, which is particularly relevant for low-thrust platforms. Low-thrust propulsion technologies such as Electric Propulsion (EP) have become popular as the primary means of propulsion for many space missions

for its fuel efficiency, which is ten times higher than conventional chemical thrusters [16]. Our goal with low-thrust engines is to determine a continuous spacecraft trajectory that satisfies initial and terminal conditions, along two different orbits, while minimizing fuel expenditure and/or Time-Of-Flight (TOF). In spite of the benefits in fuel cost, the use of low-thrust electric propulsion systems brings new challenges to mission planning and trajectory optimization. As low-thrust maneuvers require continuous control history to properly define thruster direction at each instant during the maneuver, a two-point boundary value problem is often formulated by adding control variables.

Low-thrust trajectory optimization has been investigated for many years. Moreover, in this paper, we compare solutions using two main categories: indirect and direct control methods using a Nonlinear Optimization Solver (NLOPT) and the Hp-adaptive Gaussian Quadrature Collocation and Sparse Nonlinear Programming Software (GPOPS-II), respectively. The main difference between these two methods is that while indirect methods provide highly accurate solutions, they are more sensitive to the initial guess of costate variables, which makes these methods only suitable for systems of relatively low dimensions [16]. On the other hand, direct methods transcribe the continuous-time optimal control problem into a static Nonlinear Program (NLP) which is solved by a nonlinear optimizer such as the Interior Point Optimizer (IPOPT) [18] or the Sparse Nonlinear Optimizer (SNOPT) [14]. The NLP in GPOPS-II relaxes the sensitivity issue of the initial guess by using an hp-adaptive Legrendre-Gauss-Radau pseudospectral collocation, and parameterizing both the state and control using this orthogonal collocation strategy.

The proposed optimization EPIC architecture is tested on various orbital scenarios: low-thrust transfers, chaser satellite in a Natural Motion Circumnavigation (NMC) orbit and rendezvous/flybys with some obstacle constrains targeted to proximity operations in LEO. Fig. 1 shows the overall CONOPS for the EPIC project.

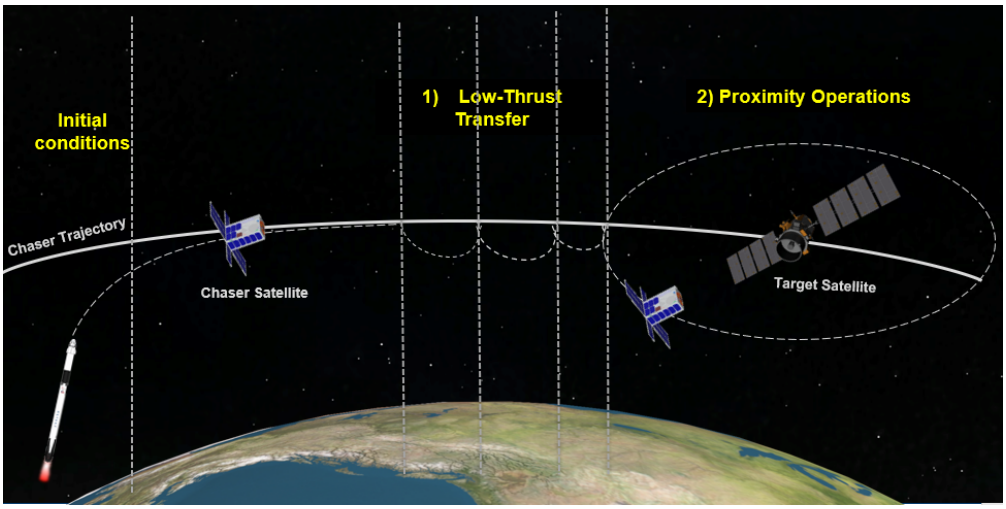

Fig. 1: EPIC project (CONOPS): low-thrust transfers and proximity operations for satellite inspection.

Because of the limited amount of fuel capacity, such maneuvers should be planned as efficiently as possible while meeting mission requirements. Increased satellite density in the LEO regime calls for the development of autonomous operations in space including autonomous safety of flight. A method for safety analysis is using barrier functions [2], which is based on encoding information about long-term safety into local information about a scalar function. Barrier functions can be used to guarantee the forward invariance of a subset of the dynamical system's state space; unfortunately, barrier functions are notoriously difficult to synthesize analytically for systems with complex dynamics or subject to uncertain disturbances. As a result of that, recent trends in safety analysis have included the use of neural networks for learning Lyapunov and barrier certificate functions [1, 6].

This paper is organized as follows: Section 2 illustrates the analytical expressions that were considered during the indirect/direct comparison of low-thrust transfers approximations. Section 3 describes the EPIC software architecture, Section 4 shows the optimal trajectories for each case a) low-thrust transfer b) NMC and c) rendezvous and flybys operations using GPOPS-II. Section 5 introduces the safety analysis for satellite inspection, and Section 6 summarizes the main findings of this research and provides insight for future work.

2. OPTIMIZATION METHODS

We describe the mathematics for both indirect and direct methods for proximity operations between satellites in the following subsections.

2.1 Indirect Method

One method to solve optimal control problems can be solved by applying the calculus of variations, obtaining the first-order necessary conditions, and solving the resulting boundary value problem. We approach this method by implementing, Kechichian [21, 15] and Feistel [8] which is a derived a solution in equinoctial coordinates to obtain thrust control to get from an initial orbit to a final orbit using minimal fuel and time.

The classical solution to this problem is usually expressed in terms of the orbital elements (a, e, w, ω, Ω, v), which are the the semi-major axis, eccentricity, inclination, argument of periapsis, right ascension of the ascending node, and true anomaly. When nonconservative perturbations such as thrusting are involved, Gauss's planetaty equations are used. Using the variational equations with classical elements brings ambiguities for circular and equatorial orbits (i.e., e=0, i=0). As a consequence, equinoctial elements or modified equinoctial elements can be used to overcome such singularities. For our indirect implementation via the NLOPT nonlinear solver, we used the following set of equations for the equinoctial elements:

$$\begin{aligned} h &= e\sin(\omega+\Omega) \\ k &= e\cos(\omega+\Omega) \\ p &= \tan\left(\frac{i}{2}\right)\sin(\Omega) \\ q &= \tan\left(\frac{i}{2}\right)\cos(\Omega) \\ \lambda &= \Omega+\omega+M \end{aligned} \tag{1}$$

The rate equation describing the system constraints in inertial, equinoctial coordinates is

$$\dot{z} = c + B_L(u+d) \tag{2}$$

where c and B_L are the vectors of the matrix of the equation of motion which transforms accelerations in Euler-Hill coordinates to equinoctial elements, u is the thrust perturbation acceleration and d is the natural perturbation acceleration vector. The optimal thrust control requires an augmented Hamiltonian that combines the Lagrange multipliers with the system constraints as shown below:

$$H = \lambda^T\left[c + BL(u+d)\right] \tag{3}$$

The vector of Lagrange multipliers is defined as

$$\Lambda = [\lambda_a\ \lambda_h\ \lambda_k\ \ \lambda_p\ \lambda_q\ \lambda_L] \tag{4}$$

As shown in equation (4), there is a multiplier for each of the equinoctial orbital elements. Using the defined Hamiltonian, the equation describing the evolution of the Lagrange multipliers or adjoint variables for the fuel-optimal trajectory is

$$\dot{\lambda} = -\left(\frac{dH}{dz}\right)^T = -\left[\frac{dc}{dz} + \frac{d(BL(u+d))}{dz}\right]^T\lambda \tag{5}$$

Therefore, the objective here is to minimize for fuel usage such that H=0

$$u* = \underset{u \in \cup}{\operatorname{argmin}} H \tag{6}$$

$$\psi(X(t_0), t_0, X(t_f), t_f) = 0 \tag{7}$$

where $\cup$ is the feasible control set and X is the state vector. We subject the dynamics to path constraints and minimize fuel usage u by setting up the states and costate equations for each of the Lagrange multipliers represented by

$$\lambda(t_0) = \frac{\partial \phi}{\partial X(t_0)} + \Lambda^T \frac{\partial \Psi}{\partial X(t_0)} \tag{8}$$

$$\lambda(t_f) = \frac{\partial \phi}{\partial X(t_0)} - \Lambda^T \frac{\partial \Psi}{\partial X(t_f)} \tag{9}$$

where ϕ is the function of terminal conditions (i.e., $\Phi = \phi + \Lambda^T \Psi$), Ψ is the terminal constraint vector and Λ is the vector containing Lagrange multipliers.

Therefore, Hamiltonian equation for both state and costates become:

$$H(t_0) = \frac{\partial \phi}{\partial X(t_0)} - \Lambda^T \frac{\partial \Psi}{\partial X(t_0)} \tag{10}$$

$$H(t_f) = -\frac{\partial \phi}{\partial X(t_0)} + \Lambda^T \frac{\partial \Psi}{\partial X(t_f)} \tag{11}$$

Finally, the perturbation acceleration and the partials of the acceleration with respect to the equinoctial elements were used as in [9], as shown below:

$$U = \frac{\mu}{r} \sum_{n=0}^{\infty} \sum_{m=o}^{n} \left(\frac{a_e}{r}\right)^n P_n^m(w) \left(C_{n,m} \cos(m(l_{\text{GEO}})) + S_{n,m} \sin(m l_{\text{GEO}})\right) \tag{12}$$

where μ is the gravitational constant for the central body, e, the equatorial radius of the central body, r, the instantaneous orbital radius, w, the sine of the declination of the satellite, l_{GEO} the longitude of the satellite in the body-fixed frame, and $C_{n,m}$ and $S_{n,m}$, the spherical harmonics of the Legendre polynomials.

2.2 Direct Method

For the direct method formulation, we used the GPOPS-II software. We consider here a rendezvous problem where a chaser satellite is flying in relative motion with respect to a target spacecraft. The goal is to execute a continuous guidance command that drives the chaser satellite from an initial position to enter an NMC orbit relative to the target. More specifically, we are interested in numerically finding open-loop solutions that minimize a specified cost (e.g. fuel expenditure, time of flight, etc.). Consequently, we focus on the minimum-fuel constrained rendezvous problem, which is defined as follows

$$\begin{aligned} \min_{t_f, T} \quad & \text{T} \\ \text{s.t.} \quad & x(t_0) = x_{start}, \\ & x_(t_f) = x_{t_{final}}, \\ & \phi_i(x(t), t) = 0 \qquad i = 1, 2, \ldots N \end{aligned} \tag{13}$$

Where $\phi(x(t),t)$ represent the path and Time-Of-Flight (TOF) constraints that are necessary to compute fuel-efficient paths considering obstacles during autonomous proximity operations [11]. For this study, we considered the Clohessy-Wiltshire (CWH) equations using RSW frame. CWH equations are described below:

$$\begin{aligned} \ddot{x} - x\left(\dot{\theta} + 2\frac{\mu}{r_c^3}\right) - y\ddot{\theta} - 2\dot{y}\dot{\theta} &= 0 \\ \ddot{y} + x\dot{\theta} + 2\dot{x}\dot{\theta} - y\left(\dot{\theta}^2 - \frac{\mu}{r_c^3}\right) &= 0 \\ \ddot{z} + \frac{\mu}{r_c^3} z &= 0 \end{aligned} \tag{14}$$

where $\ddot{x}$, $\ddot{y}$ and $\ddot{z}$ are the satellite acceleration, $\ddot{\theta}$ is the angular acceleration expressed in true latitude [schaub], μ is the Earth's gravitational constant, $3.986 * 10^{14}\ m^3/s^2$, and r_c is the orbit radius of the primary spacecraft. A detailed derivation of the solution to the CWH equations (i.e., with a_x,a_y, and $a_z = 0$), can be found in [20].

The optimal control problem (OCP) is transcribed as an NLP using IPOPT with RPM-differentiation as the setup method. The perturbation acceleration model and the partials of the acceleration with respect to modified equinoctial elements are implemented as in the GPOPS-II low-thrust code guide example [3].

3. EPIC FRAMEWORK

The EPIC software architecture in Fig. 2 consists of analyzing several phases of satellite motion aimed for satellite inspection in LEO. We use primarily the GPOPS-II optimization package in MATLAB to calculate both propagation and computational times, delta-Vs and feasibility of path constraints for orbital scenarios. Depending on the optimization phase, a propagator is used (i.e., in the Earth-Centered Inertial (ECI) frame or in the Radial, Along-track, Cross-track (RSW) frame) to describe the relative motion between satellites or Earth. For each phase, we create an input containing the objective function, lower and and upper bounds and path constraints. Such input is passed as a structure to the proprietary GPOPS-II pseudospectral discretization method, solving the optimal control problem as an NLP via IPOPT or SNOPT. GPOPS-II also provides a mesh refinement to generate high accurate solutions. Finally, a neural networks algorithm is embedded in this platform as a verification procedure to guarantee safety in proximity operations.

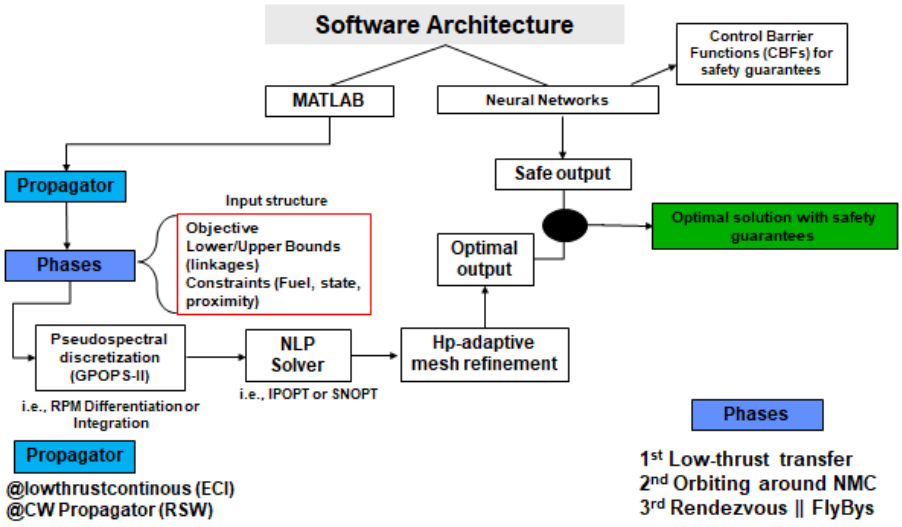

Fig. 2: EPIC software architecture: The structure of the various phases is taken as input in the GPOPS-II discretizer. Control Barrier Functions (CBFs) are synthesized to update the control output for safety guarantees of proximity operations.

4. CASE STUDIES

We divide this section into several phases. The goal of the first phase is to inject the chaser satellite into and maintenance of relative motion proximity operations trajectories such as natural motion circumnavigation (NMC) with respect to a target satellite. The second phase is simply the resulting natural motion of the terminal conditions from phase one for a specified amount of minimum dwell time. The third phase starts at the terminal conditions of phase two, with the goal of performing rendezvous and flybys with some boundary conditions. We assume that the chaser starts at an arbitrary initial condition nearby target for the CWH equations to apply. Hence, 3 cases are studied here: a) low-thrust transfer to enter an NMC b) orbiting in an NMC and c) rendezvous and flybys operations.We also compare the low-thrust transfer GPOPS-II solution with an indirect method solution using the NLOPT solver from [9].

4.1 Low-Thrust Transfers

The goal of the first optimization segment of EPIC is to compute the optimal trajectory such that a chaser satellite meet a target satellite at some point in time before entering an NMC orbit. Fig. 3 shows a graphic representation of this goal (RSW frame).

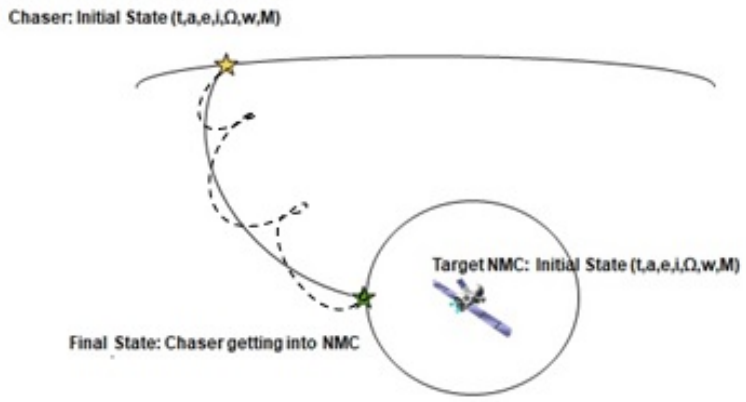

Fig. 3: The goal is to calculate chaser satellite's trajectory meeting a target satellite at some point in time and delta-V. We assume that target satellite is [1-10deg] ahead from the chaser satellite in mean anomaly.

We propagate both the chaser and target satellite at similar initial conditions (i.e., varying only the mean anomaly from 0 to 10 deg off the nominal case, which is 30 deg) and calculating the propagation time and delta-Vs for various specific impulses, Isp, (i.e., 220 s for chemical and 6000 s for ion engines) and thrust levels (i.e., 1 N and 4 mN), respectively. Table 1 shows the initial parameters of the propagation.

Table 1: Initial Orbital Parameters

Keplerian	Chaser Satellite	Target Satellite
Semi-major axis (km), a	6800.0	6800.0
Eccentricity, e	0.001	0.001
Inclination (deg), i	97.7	97.7
Right ascension ascending node (deg), Ω	10.0	10.0
Argument of perigee (deg), ω	90.0	90.0
Mean anomaly (deg), M	30.0	30.0-40.0

As expected, Fig. 4 shows that low-thrust forces consume much lower delta-V than high-thrust forces as we increase the mean anomaly difference initial conditions for both spacecraft. However, opposite behavior is found for the propagation time that takes to complete the transfer.

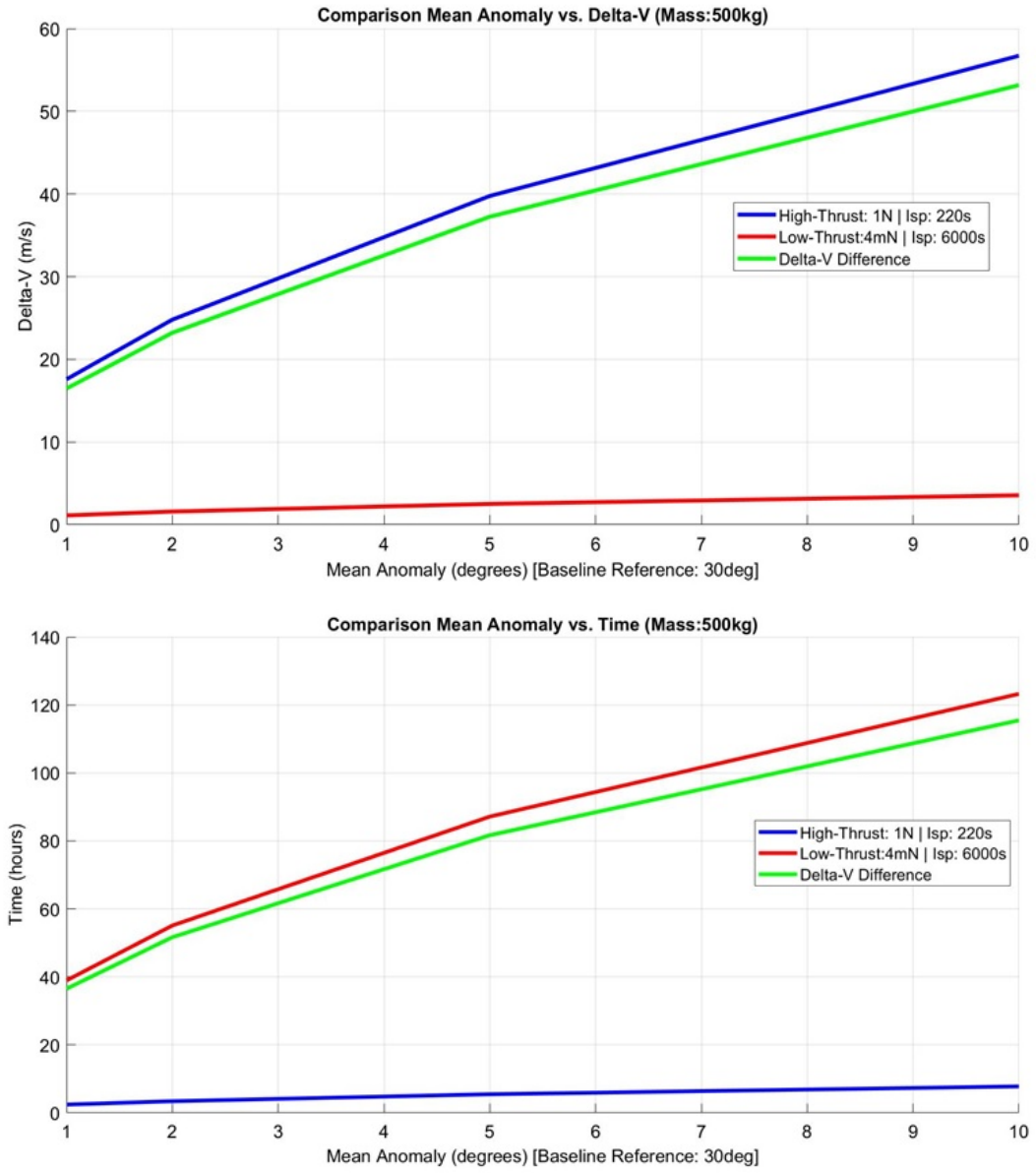

Fig. 4: Calculated high-thrust and low-thrust forces for chaser maneuvers (i.e., varying mean anomaly offset of the target satellite (1-10deg) from chaser satellite which is at 30deg). (Top) plot represents calculated delta-V while varying mean anomaly. (Bottom) plot represents calculated time while varying mean anomaly.

We analyze the behavior of the chaser satellite's orbital elements during the low-thrust transfer and Fig. 5 represents the variation of orbital elements for a difference in 10 deg of mean anomaly of the target satellite. Argument of perigee is a free variable in this set up, as we aim to get chaser into the NMC orbit with the minimum necessary fuel. However, for the two-body orbit, argument of perigee remains fixed. Fig. 5 shows that the chaser reduces the altitude to speed up and catch up with the target satellite. Other orbital elements remain fixed as intended for the terminal conditions of this optimal control problem.

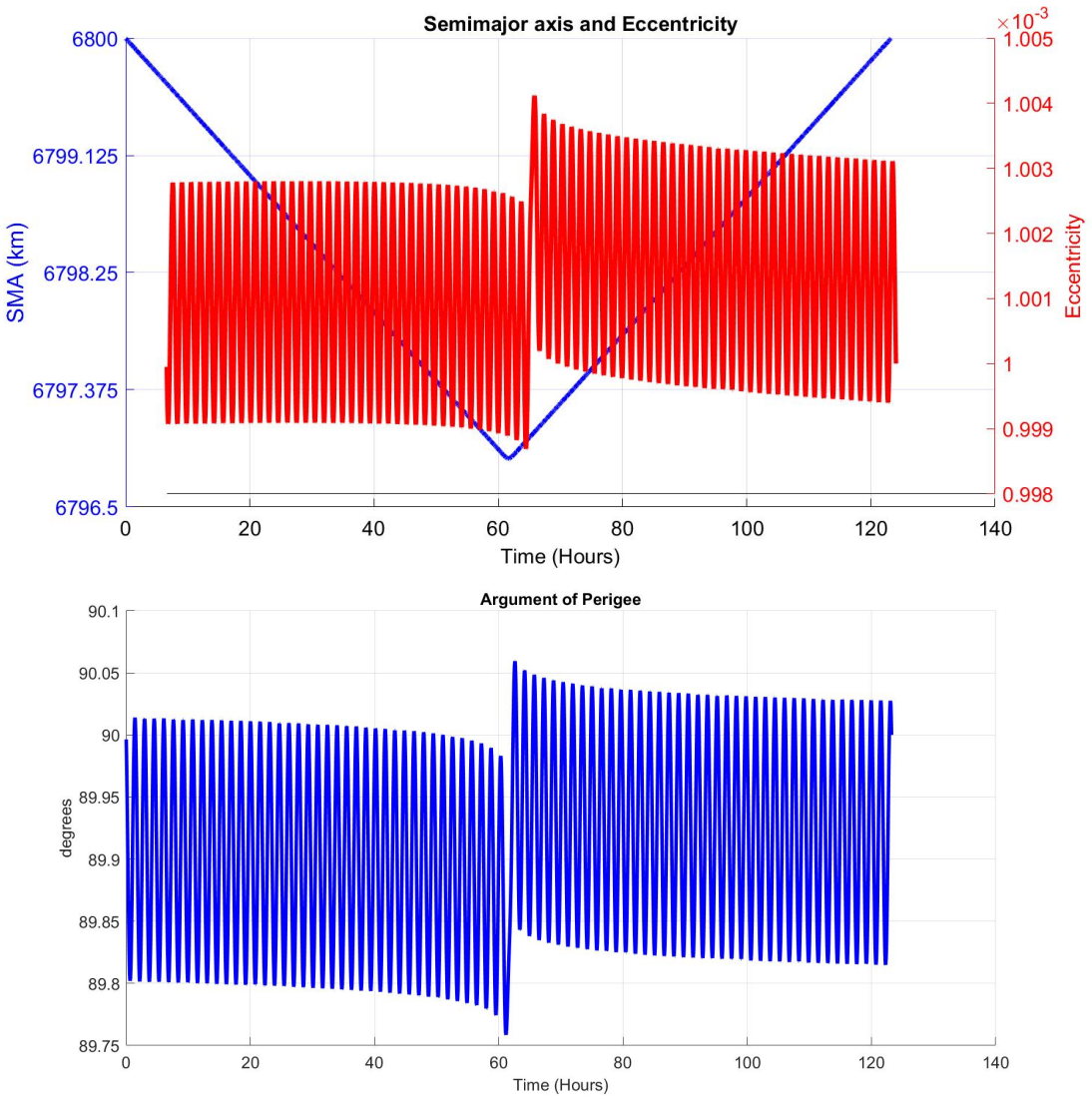

Fig. 5: Orbital elements of the chaser satellite during the low-thrust transfer: Semi-Major Axis and Eccentricity (Left), Argument of Perigee (Right). Inclination and RAAN remained constant.

Implementing the same procedure via NLOPT yields in the similar results but a less computational time duration. The main advantage of GPOPS-II over NLOPT is the reduction of complexity in merging phases within multi-path optimization schemes. Furthermore, co-state equations are not derived due to the direct formulation NLP transcription mechanism via pseudospectral methods. In GPOPS-II, we used RPM-Automatic Differentiation and IPOPT to solve for the low-thrust transfer, the default GPOPS-II tolerances were used, and no special treatment regarding the settings were required to obtain the solutions. Computations were performed on an Intel-Core i5 processor for both methods. Table 2 shows the main results between NLOPT vs. GPOPS-II for low-thrust transfers. We assume Isp of 6000 s and continuous thrust levels of 4 mN.

Table 2: NLOPT and GPOPS-II tradeoffs

Software	Computational Time	Total delta-V	Transfer Time
GPOPS-II- IPOPT	1 min 3.32 s	5.1 m/s	115.1 hr
NLOPT Solver	0.7 s	3.55 m/s	130.3 hr

These results does not always agree in transfer time or delta-Vs. This problem is currently being investigated and we suspect it might be an issue in the sparse representation of the optimal trajectory returned by GPOPS-II. Overall, we found that the OCP transcription into an NLP via GPOPS-II is easier as the multi-segment optimization structure is already built-in. However, the package requires a paid license while the NLOPT solver is free. For the remaining sections of this paper, we only use GPOPS-II/IPOPT due to the capability of linking various phases without the nested loops involved using the current interface with the NLOPT solver.

The thrust and Isp levels for the low-thrust transfer using ion engines are also shown in Fig. 6. The total delta-V for this transfer is 3.55 m/s where most of the maneuver is performed through the in-track component. In-track acceleration is $-0.8e-5\ m/s^2$ for the first half of the transfer, then it switches to $+0.8e-5\ m/s^2$ for the second half of the transfer. Isp and thrust levels remained constant during the transfer which becomes useful when doing comparison with thrust/Isp levels of chemical propulsion.

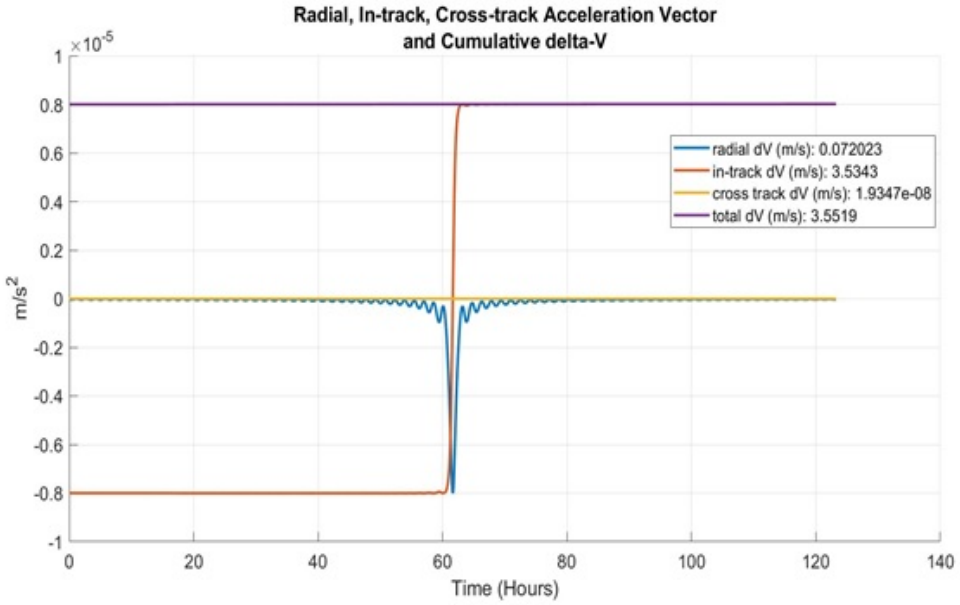

Fig. 6: Cumulative delta-V in the Radial, In-track, Cross-track acceleration vectors of this satellite

4.2 Orbit around an NMC

Phase two represents the natural, unforced motion due to the terminal conditions from phase one. The motion is a uses two-body motion in the RSW frame. The dynamics are simply the left hand side of Eq. (14) set equal to zero and propagated forward numerically using the Clohessy-Wiltshire Hill Linearize equation of motion. We set final time of propagation as 86,400 seconds or one-day.

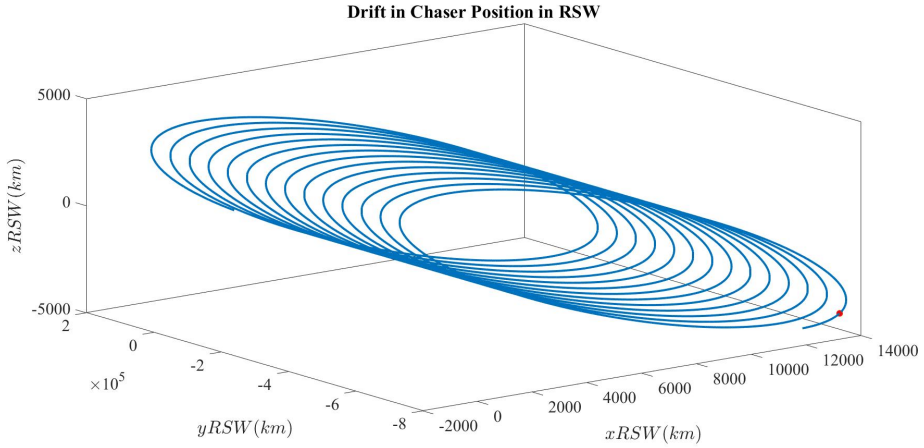

Fig. 7: chaser's position due to two-body natural motion. Propagation in perfomed in RSW during one-week propagation.

Now that the target satellite has entered an NMC, proximity operations can be analyzed. For example, delta-Vs required for rendezvous and flyby missions and safety verification aimed to satellite inspection.

4.3 Rendezvous/Flybys

We perform rendezvous and flybys operations to get near the origin considering a keep out region located at 10 km from the origin. The optimization set up combines state and input vector solutions to minimize the objective function in the Bolza form. The optimization framework is described as shown in Eq. (13) via GPOPS-II. Quadratic cost for implementing path constrains is described below:

$$\begin{aligned} \min_{J} \quad & J = \tfrac{1}{2}\int_{t}^{\infty} \mathrm{X}^{T}(t) R_{zz} \mathrm{X(t)} + \mathrm{U}^{T}(t) R_{uu} U(t) \\ \text{s.t.} \quad & \dot{X} = AX(t) + BU(t) \end{aligned} \tag{15}$$

where A and B are the state dynamics and control input (i.e., time-invariant but time-varying dynamics are also possible), respectively, and $U(t)$ and $X(t)$ represent the actuators and state, respectively. We assume perfect state knowledge of the system. Derivation on A and B matrices can be found on here [10].

R_{zz} matrix represents the weights applied to the states. If the components of the R_{zz} matrix have larger absolute value, the controller will work harder to fix the state errors. The R_{uu} matrix represents the weight applied to the control input. If the components of the R_{uu} matrix have larger absolute value, the controller will minimize control input at the expense of state error. R_{zz} and R_{uu} matrices are defined using Bryson's rule [5].

$$R_{uu} = \begin{bmatrix} \frac{\alpha_1^2}{x_{1max}} & 0 & 0 \\ 0 & \frac{\alpha_2^2}{x_{2max}} & 0 \\ 0 & 0 & \frac{\alpha_3^2}{x_{3max}} \end{bmatrix}; \qquad R_{xx} = \begin{bmatrix} \frac{\phi\beta_1^2}{u_{xmax}^2} & 0 & 0 \\ 0 & \frac{\phi\beta_2^2}{u_{ymax}^2} & 0 \\ 0 & 0 & \frac{\phi\beta_3^2}{u_{zmax}^2} \end{bmatrix}; \tag{16}$$

where x_{1max}, x_{2max} and x_{3max} refer to the maximum altitude and along-track delay states that are expected, α_1, α_2 and α_3 represent the weights on the states, β_1 represents the weight on the control input, and ϕ represents the weight between the states and the control input. The maximum control inputs, u_{xmax}, u_{ymax}, and u_{zmax}, vary depending on the engine, so a variety of thrusts for low-thrust engines are considered from [5]. The thrust is divided by the satellite mass to determine the acceleration of the engines. These values serve as an initial guess for these weights. However, more tuning is needed during the simulation based on the behavior found. Table 3 shows the Bryson's rule values for this example.

Table 3: Bryson's rule values

Variable	Altitude	Along-Track	Unit
x_{1max}	112.75	112.75	m
x_{2max}	4.6635	4.6635	m
x_{3max}	4.6635	4.663	s
α_1	100	1	-
α_2	1	100	-
α_3	1	1	-
ϕ	10	10	-
β_1	10	10	-

If our objective is to penalize only for fuel usage, we can generate the following trajectories for position and velocity of our chaser satellite. Flybys with path constraints and higher fidelity models will be shown in the journal article being submitted this fall.

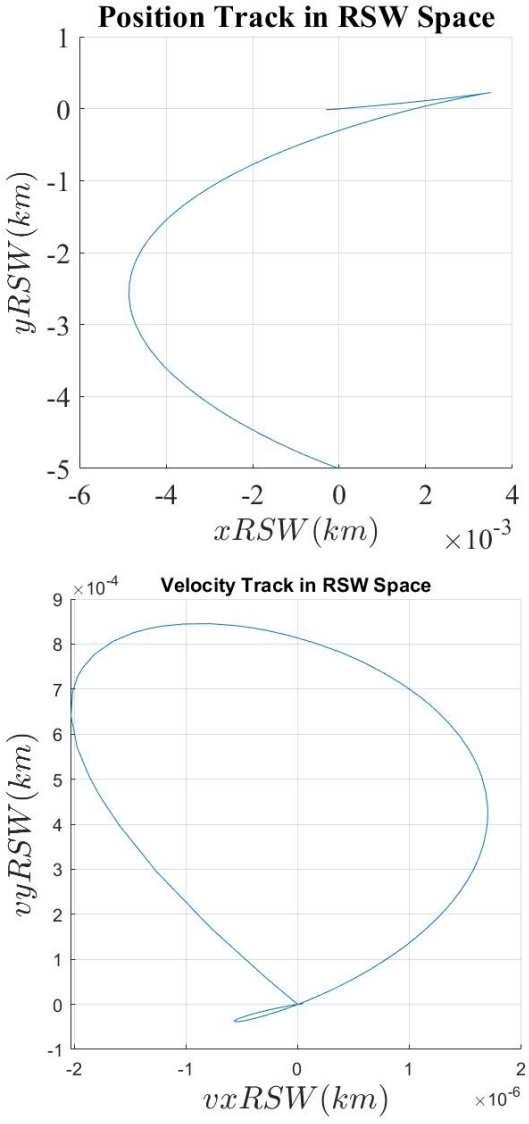

Fig. 8: Optimal trajectory calculated for chaser satellite in RSW frame (2-D Plot). (Right) plot represents the position track and (left) plot represents the velocity track.

5. SAFETY ANALYSIS

Original application of this safety verification procedure can be found in Garcia et al. [10] for a Model Predictive Controller (MPC), ideal for real-time operations. For spacecraft conducting proximity operations, it is important that the target spacecraft remain near the target satellite without getting close enough to cause a collision. In the RSW frame, the target satellite is at the origin, and so we can express the safety requirement as a constraint on the state of the chaser spacecraft:

$$x_{min} \leq ||\mathbf{x}|| \leq x_{max} \tag{17}$$

for the state $\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]$, keep-away minimum norm x_{min} and keep-in maximum norm x_{max}. This constraint is illustrated in Fig. 9.

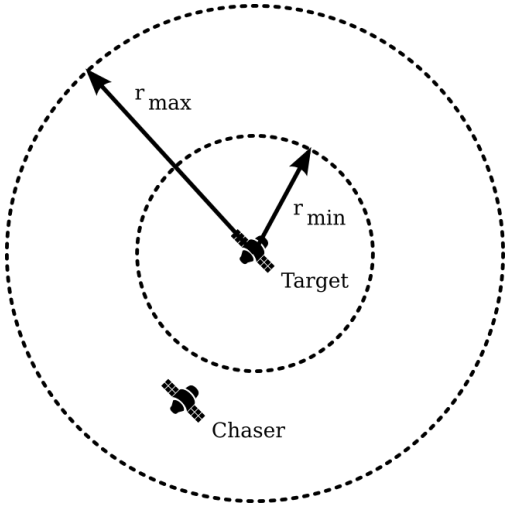

Fig. 9: The chaser satellite must remain between a minimum and maximum distance from the target satellite.

Verifying that a spacecraft controller satisfies this safety constraint is challenging for a number of reasons:

1. The RSW frame is convenient for expressing the safety constraint (17), but it relies on a linearization of orbital dynamics and thus does not capture higher-order nonlinear effects. The safety verification process should account for these effects.
2. The chaser spacecraft may not be static; it will often have its own control policy, which may not be known ahead of time (or at all). The safety verification process should be robust to some (bounded) maneuvering by the chaser spacecraft.
3. Designing a spacecraft controller is challenging, so the safety verification process should allow maximum flexibility in designing the controller.

A safety verification strategy that meets all three requirements is that of robust Control Barrier Functions (CBFs). A CBF is a scalar function of state $h : R^n \mapsto R$ such that $h(\mathbf{x}) < 0$ in the safe region, $h(\mathbf{x}) > 0$ in the unsafe region, and the time derivative of h satisfies the inequality

$$\inf_u \left[\frac{dh}{dt}\right] = \inf_u [\nabla h \mathbf{f}(x) + \nabla h \mathbf{g}(\mathbf{x})\mathbf{u}] \leq -\lambda h(\mathbf{x}) \tag{18}$$

for control-affine dynamics, $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}$. If h satisfies these requirements, then established results in control theory imply that there exists a family of feedback control policies that cause the state to remain in the safe region for all future time [2]. Even if these conditions are violated in some regions, as long as they hold throughout the region where $h < 0$ the safety guarantees hold and safe controllers are guaranteed to exist. In particular, this family of safe controllers

is all Lipschitz continuous control policies taking values from the set $K_{CBF} = \{\nabla h\mathbf{f}(x) + \nabla h\mathbf{g}(\mathbf{x})\mathbf{u} \leq -\lambda h(\mathbf{x})\}$. An arbitrary Lipschitz controller $\pi_{baseline}$ can be projected onto this safe set by solving the CBF quadratic program (CBF-QP)

$$\begin{aligned} &\min_{u} ||u - \pi_{baseline}||^2 \\ &\text{s.t. } \nabla h\mathbf{f}(x) + \nabla h\mathbf{g}(\mathbf{x})\mathbf{u} \leq -\lambda h(\mathbf{x}) \end{aligned} \tag{19}$$

This CBF-QP can act as a safety filter, satisfying requirement (1) by allowing the system designers to choose any $\pi_{baseline}$ they wish, so long as it is filtered to a safe policy by solving (19). However, in order to meet requirements (2) and (3), we must employ a CBF that is robust to uncertainty and non-cooperative disturbances in the inputs. To do this, we employ a robust certificate as discussed in [7].

Our safety analysis must be robust to two sources of disturbance: higher-order effects ignored in the linearized CWH dynamics and worst-case movement from the target spacecraft. We assume that the worst-case disturbance from the sum of these two effects is bounded along each axis, i.e. $d_x, d_y, d_z \in [d_{min}, d_{max}]$. For the following analysis, we take $d_{max} = -d_{min} = 0.01$. By exploiting the fact that condition (18) is affine in these uncertain disturbances, if (18) holds for all worst-case scenarios $\mathbf{d} = (d_x, d_y, d_z) \in \{d_{min}, d_{max}\}^3$ then it is guaranteed to hold for any scenario in the convex hull of those extreme points [7]. Using this fact, we can augment the CBF-QP in (19) to construct a robust version, which is sufficient to enforce safety despite perturbations from nonlinear effects and potentially non-cooperative actions by the target satellite:

$$\begin{aligned} &\min_{u} ||u - \pi_{baseline}||^2 \\ &\text{s.t. } \nabla h(\mathbf{f}(x) + \mathbf{d}_i) + \nabla h\mathbf{g}(\mathbf{x})\mathbf{u} \leq -\lambda h(\mathbf{x}) \qquad i = 1, 2, \ldots 8 \end{aligned} \tag{20}$$

This robust CBF-QP can be used as a safety filter, guaranteeing that the system remains safe despite perturbations. The filtering architecture is shown in Fig. 10: the CBF-QP projects the baseline control policy onto the set of safe controllers, allowing it to be used with baseline controllers that are difficult to rigorously verify (such as MPC and LQR controllers). This filtering architecture provides flexibility and robustness, satisfying our three requirements for safety verification; however, there are three notable drawbacks.

First, this architecture shifts responsibility for safety to an online monitor in the form of the robust CBF-QP. Instead of using a more accurate perturbation model and propagating the dynamics forward in time, the CBF-QP assumes that an accurate state estimate will allow it to adapt disturbances online. The feasibility of the CBF-QP can be verified offline, but the need for an accurate, high-frequency state estimate is a hard requirement for this approach.

Second, the CBF-QP safety filter has no intrinsic notion of optimality (other than closeness to the baseline controller). By design, the CBF-QP is concerned only with maintaining safety. If a mission requires optimal use of resources or positioning relative to the target, those considerations are delegated to the baseline controller. For example, a controller may be used to ensure low propellant use, while the CBF-QP is used to filter the output of the algorithm.

Finally, robust certificate functions like CBFs are notoriously difficult to synthesize by hand [13]. Fortunately, recent years have seen the development of learning-based methods for synthesizing CBFs [4, 1, 6, 19]. In this work, we draw on the approach in [7] for learning robust control barrier functions using neural networks. We train a neural network with 4 hidden layers of 128 tanh activation units each, using stochastic gradient descent to minimize the violation of the constraints (20) in the robust CBF-QP. We train the CBF network on 100000 data points sampled uniformly from $[x, y, z, \dot{x}, \dot{y}, \dot{z}] \in [-2.5, 2.5]^3 \times [-1, 1]^3$, compute the violation of constraints (20) on each datapoint. Since some flexibility is needed in determining the shape of the robust CBF, we modify the safety constraints in Fig. 9 to specify safe and unsafe regions of state space, leaving some separation between these sets (states in this buffer zone may be either safe or unsafe).

A contour plot of the trained CBF network is shown in Fig. 11, including the zero-sublevel set marking the invariant region (outlined in blue). Trajectories starting at states within this zero-sublevel set are guaranteed to remain within that set when controlled using the robust CBF-QP. The blue contours match the safety specifications from Fig. 9,

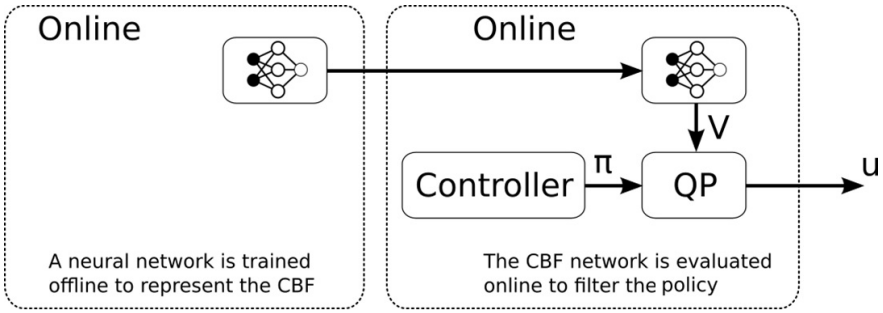

Fig. 10: A control barrier function (CBF) can act as a filter for any control policy $\pi(x)$ (in our case, GPOPS-II), endowing the combined controller with guarantees on long-term safety.

shown in pink and green. The conditions (20) were found to be satisfied at all points in Fig. 11 within the blue zero-sublevel set; these conditions were checked on a uniform grid of points with maximum spacing 0.01 between points. We verified the validity of this robust CBF by checking the conditions (20) were satisfiable at all points within the blue invariant zero-sublevel set on a uniform grid with maximum spacing 0.01 between adjacent points — we found no violation of the CBF conditions on this set. Because the CBF conditions are satisfied throughout this blue set, this set is guaranteed to be invariant. The region where the CBF conditions is not satisfied and must be relaxed is shown shaded in silver; note that the region where relaxations are required is outside the blue sublevel set.

From Fig. 11, we see that the CBF certificate certifies an invariant region (blue) that does not fully overlap with the specified safe region (green). Ideally, the area between the blue curves should be a superset of the area between the green curves, and it should have no overlap with the unsafe region (inside the inner pink circle or outside the outer pink circle). By checking the certificate values on a uniform grid as described above, we find that our barrier certificate certifies 99.52% of the desired safe region as forward invariant, and the overlap between the certified invariant region and the specified unsafe region is 0%.

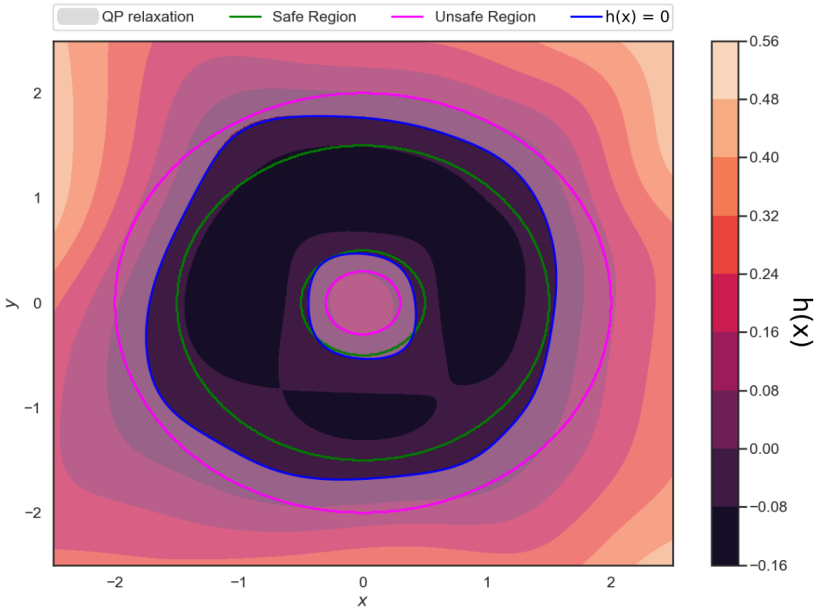

Fig. 11: The robust CBF synthesized using a neural network. The blue contour indicates the region that the CBF guarantees to be forward invariant: all states starting between the blue curves will remain between the blue curves indefinitely.

6. CONCLUSION

Trajectories were optimized under certain constraints for satellite inspection in the LEO regime. GPOPS-II set up is more convenient for multi-segment optimization and its convergence is not as sensitive as NLOPT to the initial guesses needed to start the nonlinear solvers. However, computational time with GPOPS-II takes longer than NLOPT, especially for high-fidelity models. An innovative approach to guarantee safety for these controllers was presented for various applications including safety verification in collision avoidance operations. In addition, we synthesize robust control barrier functions using neural networks, which provides safety-aware filtering for the control policies presented here, guaranteeing long-term safety despite perturbations.

The presented safety verification method has strong potential for application to safety verification of autonomous collision avoidance systems, especially those used by different spacecraft operators operating at overlapping altitudes. This approach avoids many of the proprietary issues associated with screening operator interactions using reachability based methods, does not require exquisitely accurate orbit propagation to ensure safety, and can be applied generically in a controller-agnostic manner without requiring extensive customization for each user or control algorithm. Nevertheless, this approach is reliant on the mutual sharing of timely accurate decision-quality satellite ephemeris for relevant spacecraft.

7. FUTURE WORK

We plan to add more fidelity to the current EPIC architecture for the various controllers presented here by adding more perturbation effects, and scaling our trajectory problem with moving obstacles. We also plan to add power and propulsion failures as part of our mission constraints. Wrasse propagator will be integrated with GPOPS-II instead of the low-thrust/CWH propagators used in MATLAB. Simulated trajectories, combining our controller output with the CBF-QP filter, would also show the completeness of the safety verification method as well. The goal of this architecture is to enable satellite inspection in space operations by performing rendezvous and proximity operations (RPO) and on-orbit satellite servicing (OOS) operations in LEO.

8. ACKNOWLEDGMENTS

The authors would like to acknowledge the Draper Scholars Program for providing funds to support this research. The authors would also like to thank the MIT Lincoln Laboratory for providing the GPOPS-II commercial license and internship experience through the GEM fellowship program. We would also like to acknowledge Miles Lifson, Brain Gaudet and Prof. David Arnas for their guidance implementing path constraints and linkages for multi-segment optimization in GPOPS-II. We would also like to thank Prof. Chuchu Fan and Charles Dawson from MIT AeroAstro for previous collaboration and support synthesizing CBFs.

9. REFERENCES

[1] Alessandro Abate, Daniele Ahmed, Mirco Giacobbe, and Andrea Peruffo. Formal synthesis of lyapunov neural networks. *IEEE Control Systems Letters*, 5(3):773–778, Jul 2021.

[2] Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European Control Conference, ECC 2019*, pages 3420–3431. Institute of Electrical and Electronics Engineers Inc., jun 2019.

[3] J.T. Betts. Practical methods for optimal control and estimation using nonlinear programming. *SIAM Press*, 2009.

[4] Ya Chien Chang, Nima Roohi, and Sicun Gao. Neural lyapunov control. In *Advances in Neural Information Processing Systems*, volume 32, pages 3245–3254, 2019.

[5] Mario Contreras, David Arnas, and Richard Linares. State-space controller for leo, low-thrust satellite constellations. 02 AAS/AIAA Space Flight Mechanics Conference 2020.

[6] Hongkai Dai, Benoit Landry, Marco Pavone, and Russ Tedrake. Counter-example guided synthesis of neural network lyapunov functions for piecewise linear systems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 1274–1281, 2020.

[7] Charles Dawson, Zengyi Qin, Sicun Gao, and Chuchu Fan. Safe nonlinear control using robust neural lyapunov-barrier functions, ubmitted for review on IEEE Transactions on Robotics (T-RO). Sept, 2021.
[8] Andrew Feistel. Perturbing accelerations in low thrust trajectory optimization. Technical report, University of Texas at Austin, May 2007.
[9] Z.J. Folcik. Accurate low-thrust orbit transfer solutions in equinoctial elements using an analytic representation of the geopotential. In *www.anilvrao.com/Publications/ConferencePublications/trajectorySurveyAAS.pdf.*, pages 19–770. AAS/AIAA Astrodynamics Specialist Conference, Portland, ME, August 11-15, 2019, 2019.
[10] Axel Garcia, Miles lifson Dawson, Charles, and Linares Richard. Model predictive control and safety analysis for satellite collision avoidance. *AAS*, 08 2021.
[11] Brian Gaudet, Richard Linares, and Roberto Furfaro. Adaptive guidance and integrated navigation with reinforcement meta-learning. *Acta Astronautica*, 169, 04 2020.
[12] David Gaylor and Brent Barbee. Algorithms for safe spacecraft proximity operations. *Advances in the Astronautical Sciences*, 127:133–152, 01 2007.
[13] Peter Giesl and Sigurdur Hafstein. Review on computational methods for Lyapunov functions. *Discrete and Continuous Dynamical Systems - Series B*, 20(8):2291–2331, oct 2015.
[14] Philip E. Gill, Walter Murray, and Michael A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.
[15] Jean Albert Kechichian. *Trajectory optimization using nonsingular orbital elements and true longitude*, volume Journal of Guidance, Control, and Dynamics Vol. 20 No. 5 (1997):1003-10009. Journal of Guidance, Control, and Dynamics, 1997.
[16] Mirko Leomanni, Gianni Bianchini, Andrea Garulli, Renato Quartullo, and Fabrizio Scortecci. Optimal low-thrust orbit transfers made easy: A direct approach. *Journal of Spacecraft and Rockets*, page 1–11, Jul 2021.
[17] J.H. Meub and H. Pernicka. Spacecraft proximity operations using continuous low thrust. *Advances in the Astronautical Sciences*, 140:545–564, 01 2011.
[18] E. R. Prince, R. W. Carr, and R. G. Cobb. Fuel Optimal, Finite Thrust Guidance Methods to Circumnavigate with Lighting Constraints. In S. Ryan, editor, *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, page 110, January 2017.
[19] Zengyi Qin, Kaiqing Zhang, Yuxiao Chen, Jingkai Chen, and Chuchu Fan. Learning safe multi-agent control with decentralized neural barrier certificates, 2021.
[20] David A Vallado. "long-term numerical propagation for earth orbiting satellites. In *American Astronautical Society (AAS)*.
[21] Xiao-Li Xu. *Inclusion of higher order harmonics in the modeling of optimal low-thrust orbit transfer*, volume Vol. 56, No.1, 2008, pp.41-70. The journal of the Astronautical Sciences, 2008.