

Discrete-Time Optimization and Safe-Trajectory Generation for Satellite Formation Flying and Proximity Operations

Kristen Tetreault¹, Ian Elliott², Shane D. Ross³, and Jonathan Black⁴
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061

We present a real-time guidance and control algorithm for the generation of collision-safe trajectories of a CubeSat in proximity operations around a non-maneuvering resident space object orbiting a large body. Two discrete-time guidance and control methods are demonstrated. The first method is an open-loop finite maneuver solver for explicit time-of-flight rendezvous maneuvers calculated in non-real-time. The second method uses model predictive control for extremely close range proximity operations with safe trajectory planning calculated in better-than real-time. Both guidance methods use mixed-integer linear programming to control the relative translational movement of a CubeSat in proximity operations while simultaneously maintaining a desired three-axis orientation of the CubeSat. Example scenarios are simulated to demonstrate the proximity operations capabilities of the guidance algorithm on a 6U CubeSat. A point-to-point maneuver is simulated both with and without targeted attitude pointing, and next, with targeted attitude pointing as the CubeSat receives new information about the orientation of the reference satellite. Finally, a major rendezvous maneuver is demonstrated with proximity operations to simulate the performance of the algorithm to control a CubeSat docking with a larger satellite.

¹ Ph.D. Student, Engineering Mechanics Program, Virginia Tech 24061

² Graduate Student, Ann and H.J. Smead Aerospace Engineering Sciences, University of Colorado Boulder 80309

³ Professor, Engineering Mechanics Program, Virginia Tech 24061

⁴ Professor, Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech 24061

Nomenclature

$()_{lb}$	= minimum bounding volume value
$()_{ub}$	= maximum bounding volume value
α	= weighting factor
d	= safety buffer distance
Δv	= velocity change of satellite over maneuver
dt	= time step
\mathbf{I}	= moment of inertia matrix
J	= solver cost function
k_d	= rate gain
k_p	= proportional gain
M	= torque controls vector
o_k	= collision avoidance binary variables
ω	= angular velocity
P	= arbitrarily large scalar value
p_n	= desired position of satellite
p_t	= position of satellite at end of solver horizon
$p_{t,n}$	= separation between satellite's end-of-solver and desired position
\bar{q}	= quaternion matrix
\bar{q}_N	= desired quaternions
\bar{q}_p	= proportional term for quaternions
R_N	= desired rotation matrix
T	= horizon time
TP	= time period
u_n	= control signal in the n direction
\mathbf{u}	= satellite controls vector
v_n	= desired velocity of satellite
v_t	= velocity of satellite at end of solver horizon
\mathbf{x}	= satellite state vector

I. Introduction

Achieving prolonged proximity operation abilities within the micro-class satellites requires optimizing fuel usage. Model predictive control (MPC) is an advanced control approach that can be used to optimize controls in order to serve multiple simultaneous objectives. With MPC, a satellite can be controlled to perform a specific mission task while also incorporating fuel usage into the optimization problem.

For MPC to be feasible on-board a micro-class satellite, the optimization algorithm used must be able to solve significantly sized problems in extreme succession to produce controls in real-time. Even with the most state-of-the-art desktop workstation, nonlinear optimization model predictive controllers are infeasible [1, 2]. However, mixed-integer linear programming (MILP) optimization has proven to have highly effective solvers that also allow for binary constraints [3, 4]. A typical MILP problem can be solved on a low-end consumer computer running on internal battery power, similar processing power to that on-board micro-class satellites. The application of MPC using MILP to the control of micro-class satellites comes with modeling the relative motion of two or more spacecraft in close orbital proximity to each other. Once this motion is modeled, additional, fuel-optimal satellite controls can be incorporated to control the satellite in different mission tasks.

Mixed integer linear programming requires a linear approximation of the equations of motion for satellites in proximity. In near-circular orbits, the Hill-Clohessy-Wiltshire (HCW) equations provide a simple and time-independent special case of the time-variant linear equations of relative motion (LERM) [5]. The LERM are themselves an approximation of complex nonlinear models. While current computational ability limits the optimization solver to linear models, coupling linear-controls within nonlinear propagators has demonstrated little variance for typical mission control horizons [2]. The result of this coupling is the ability to simultaneously maintain real-time controls and complex perturbations such as the effects of J_2 on relative motion. Additionally, effects from fuel mass loss are an extremely important nonlinear factor for micro-class satellites.

Multiple satellite control algorithms applied to proximity operations have been previously researched, varying in their applicability, accuracy, and mathematical approach. Schaub designed a closed-form solution for formation reconfiguration in elliptical orbits [6, 7]. Yan and Gong used a fi-

nite horizon linear-quadratic controller for long-term formation maintenance in conjunction with the Gim-Alfriend State Transition Matrix (GA-STM), however this work did not involve constraints on the controller [8]. Additionally, the controller fired continuously whereas the work described herein does not because of the bang-off-bang profile from the minimum fuel optimal control. Richards et al. used a standard linear program to minimize fuel consumption while modeling motion with the HCW equations to optimize trajectories for robotic inspection [4]. MILP was used to account for plume impingement and collision avoidance. Tillerson et al. also applied linear programming to a formation flying situation with avoidance constraints in an elliptic orbit [9]. A similar problem was studied by Mueller et al., without the contribution of J2 [10]. All three of the previously developed works assumed a fixed time-of-flight for their maneuvers.

Schouwenaars created a model for real-time safe trajectory planning of autonomous vehicles with applications to rotary vehicles [11]. Rogers' work involved using MPC satellite proximity operations tracking a priori known fuel-optimal trajectories [12]. An early version of the model described below was applied by Nastasi et al. to determine operational lifespan of CubeSats in highly-constrained relative motion around a target satellite while maintaining a non-Keplerian orientation with respect to the target [13]. Similarly, an early version of the MPC guidance algorithm was applied by Thomas et al. to analyze CubeSat sensor performance and overall CubeSat rendezvous and proximity operations (RPO) feasibility in similar, constrained radial proximity operations [14].

Roscoe et al. uses a similar process as described here to control the upcoming CubeSat Proximity Operations Demonstration (CPOD) Mission. The CPOD Mission is a NASA mission developed by Tyvek and Applied Defense Solutions consisting of two identical 3U CubeSats with interlinking communication [15]. Roscoe's model used the Gim-Alfriend state transition matrix, which models perturbations from J2 for satellite relative motion [16]. Model predictive control is used to offset the effects of drag on the relative motion of the two CubeSats.

AeroJet Rockedyne is currently working on developing two CubeSat modular propulsion systems (MPS), the AeroJet MPS-120 and MPS-130. Both systems are design to fit within a single CubeSat U, or roughly one liter of volume. The AeroJet MPS-120 is a hydrazine-based monopropellant system while the MPS-130 will use green propellant [17]. VACCO industries has already produced

a line of modular cold gas CubeSat propulsion systems. VACCO CubeSat propulsion systems have successful flight heritage on the AeroCube 4 1U CubeSat developed by the Aerospace Corporation [18]. CubeSpace currently provides CubeSat reaction wheels capable of providing up to 2.3 mNm of torque.

The capabilities presented herein provide the framework for an autonomous on-board micro-class satellite guidance and control system for use with missions requiring targeted sensor pointing in proximity operations. The ability to generate safe trajectories with constrained attitude pointing is especially important for micro-class satellites, where volume and power for RPO sensors is at a premium. Optical primary rendezvous sensors can require constant attitude pointing at the target satellite to gather relative pose data. Examples include an autonomous vehicle tasked to quickly image a manned spacecraft. The control algorithm can use a known model of the larger spacecraft to automatically generate safe trajectories. The guidance system can also be applied to the docking of micro-class satellites with larger spacecraft.

II. Model

A micro-class satellite design to perform orbital inspection requires a relatively high thrust authority while at the same time sensitivity for minute maneuvers. Cold gas thrusters and more recently hydrazine propellant has been studied for use as CubeSat standardized hardware for micro-class satellites [13, 14]. For these propulsion systems, the ability to throttle is unavailable. This unavailability is where the binary constraint capability of MILP is key in the ability to model these binary thrusts. Another critical capability of binary constraints is collision avoidance, which is only possible for mixed integer linear solvers. Binary constraints allow for the generation of controls that ensure collision-safe trajectories of specified volumes. More complex obstacle shapes for collision avoidance can be modeled with the use of appended volumes of avoidance. The Space Shuttle is a good example of one such complex obstacle, and its appended volumes of avoidance can be found in Chapter 6 of Rogers' proximity operations work [12].

A. Satellite Relative Motion

All controls are calculated in the reference satellite's radial, in-track, cross-track (RIC) frame. The RIC frame's x-axis is defined as the reference satellite's radial direction \hat{e}_r , the z-axis is defined the angular velocity direction of the reference satellite \hat{e}_h , and the y-axis \hat{e}_θ is defined as to complete the right-handed coordinate system. For eccentric orbits, \hat{e}_θ is not equivalent to the direction of the reference satellite velocity, \hat{e}_v . For simplicity, the reference satellite is referred to as the "chief", and the relative motion satellite is referred to as the "deputy". The chief can be representative of any resident space object (RSO) while the deputy herein is modeled after a CubeSat.

The attitude of the deputy relative to the chief is defined with a body-fixed frame, \hat{b} , relative to the chief's RIC frame. Figure 1 illustrates both the chief's and deputy's frames. The chief satellite is designated with c and the deputy satellite is represented as d . The nonlinear equations of relative

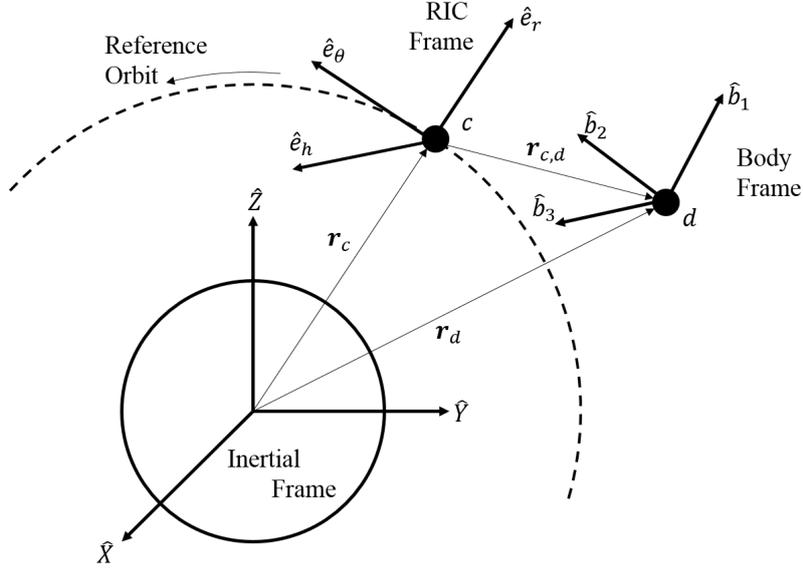


Fig. 1 Reference frame systems for two satellites in proximity operations. Bold formatting indicates a vector, i.e. r_c .

motion (NERM) for two objects orbiting in close proximity to a large central body can be linearized into the LERM [19]. In this specific case, it is valid to use the LERM instead of the NERM for a variety of reasons, though mainly the relatively short distance between the two satellites. Further discussion can be found in [5, 19] on this linearization. In the case of near-circular orbits, LERM

can be simplified further into the HCW equations described below in Eq. (1) [20].

$$\begin{aligned}\ddot{x} &= 3n^2x + 2n\dot{y} \\ \ddot{y} &= -2n\dot{x} \\ \ddot{z} &= -n^2z\end{aligned}\tag{1}$$

The motion of the deputy around the chief will remain as a relative motion ellipse during the special case of $\dot{y} = -2nx$.

B. Optimal Control Problem

The optimal control problem is set up in discrete-time form with a predefined solver horizon time, T , and simulation time step, dt . The standard mixed-integer linear program can be summarized by Eq. (2). The cost function, J , serves to define the objective of the satellite. The state of the satellite is represented as the vector \mathbf{x} . Equality constraints are used to define the effects of relative motion at each time step. Inequality constraints are used to define the maximum relative velocity the deputy can travel to balance time and fuel usage. Inequality constraints are also used to implement collision avoidance and constrain the position of the satellite to within a holding volume. Finally, inequality constraints contain the slack variables necessary for waypoint-maneuvering.

$$\text{Minimize } J(\mathbf{x}) = f^T \mathbf{x} \quad \text{Subject to: } \begin{cases} A \cdot \mathbf{x} \leq b \\ A_{eq} \cdot \mathbf{x} = b_{eq} \\ lb \leq \mathbf{x} \leq ub \\ o_k \in \{0, 1\} \end{cases}\tag{2}$$

During long-range maneuver operations, the algorithm solves for minimum-impulse solutions with a maximum impulse constraint. The magnitude of the impulse and time for each maneuver is calculated by the solver. The solver is given control over the exact number of impulses to complete the minimum-impulse maneuver. Equality constraints for position and velocity are imposed on the state of the satellite at the end of the solver horizon, p_T and v_T , respectively, which must match

the desired state, p_N and v_N .

$$J = dt \sum_n^T u_n \quad \text{Subject to: } \begin{cases} p_T = p_N \\ v_T = v_N \\ t_T = t_N \\ 0 \leq u_n \leq u_{max} \end{cases} \quad (3)$$

During MPC operations for close proximity, the solver incorporates a binary-thrust constraint to precisely dictate when, and for how long, each thruster fires. Trajectory generation during MPC includes two main phases of behavior. The first is a waypoint-to-waypoint maneuvering in which the satellite attempts to approach a designated waypoint within the designated bounding volume while using the least amount of fuel possible. The solver cost function for this behavior, $J_{waypoint}$ is described in Eq. (4) where $\Delta p_{T,N}$ is the L-1 norm of the separation of the satellite's position at the end of the solver horizon from the requested waypoint coordinate. This position separation is represented by slack variables that the solver attempts to minimize. The multi-objective cost function also includes a second term with a weighting factor α to minimize fuel-usage through minimizing controls in addition to minimizing separation to the waypoint.

$$J_{waypoint} = \Delta p_{T,N} + \alpha \sum_n^T u_n \quad \text{Subject to: } \begin{cases} v_n \leq v_{max} \\ u_n = 0, u_{max} \\ \sum o_k \leq 5 \end{cases} \quad (4)$$

The second main phase of behavior is a holding phase. During this phase, the solver no longer includes the term to minimize distance to a waypoint and only the term to minimize fuel usage remains. Typically, constraints are set to limit the position of the deputy to within a designated bounding volume relative to the chief. During a holding phase, it is advantageous to define the bounding relative volume so that the chief's collision avoidance bounds do not intersect. This prevents unnecessary computational resources dedicated to solving binary variables.

$$J_{holding} = dt \sum_n^T u_n \quad \text{Subject to: } \begin{cases} v_n \leq v_{max} \\ u_n = 0, u_{max} \\ p_{lb} \leq p_n \leq p_{ub} \end{cases} \quad (5)$$

C. Model Predictive Control

For every time step in the solver, a single set of variables listed in Eq. (6) is defined for the mixed-integer linear program. The control signals in each direction of each axis are designated u and the relative acceleration due to the equations of relative motion, $(\ddot{\cdot})$, are defined in each axis. The separation distance L-1 norm that is being minimized in an approach phase is represented as three slack variables $\Delta(\cdot)$ in each axis. These three slack variables are only defined during the final time step in the solver horizon. The collision avoidance binary variables, o_k , are defined in the following section, §D. In the holding phase, or for a scenario where there is no obstacle to avoid, the collision avoidance binary variables are not incorporated.

$$\tilde{x}_n = [\underbrace{u_x^+ \ u_x^- \ u_y^+ \ u_y^- \ u_z^+ \ u_z^-}_{\text{Control signals}} \ \underbrace{\ddot{x} \ \ddot{y} \ \ddot{z}}_{\text{Relative motion}} \ \underbrace{\Delta x \ \Delta y \ \Delta z}_{\text{Waypoint slack}} \ \underbrace{o_1 \ o_2 \ o_3 \ o_4 \ o_5 \ o_6}_{\text{Collision avoidance}}]^T \quad (6)$$

The model predictive control guidance algorithm is visualized in Figure 2. A mixed-integer linear programming solution is generated every time step of the simulation. After each run by the solver, the model predictive control algorithm propagates the thrust controls as well as the attitude controls for one time step dt . Other non-linear propagation effects can also be implemented at this point, including the mass expended by fuel-burn. The MILP solver is then run again with updated initial conditions and the same objective. This process is a finite-horizon MPC system which produces locally fuel-optimal trajectory solutions. While this form of MPC may not result in a globally optimal solution, it benefits from the ability to operate in real-time and the ability to react to new information, such as movement from the chief. The collision avoidance measures, binary throttling, and flexible maneuvering objectives makes this MPC well-suited for very short-range proximity operations as well as docking approach and hold.

D. Collision Avoidance

Mixed-integer linear programming is capable of generating collision-safe trajectories [4]. A rectangular prism bounding volume aligned with the RIC frame must be defined around the obstacle with which the trajectories must not intersect. The bounding volume is defined in the chief's RIC frame with maximum and minimum values $(\cdot)_{ub}$ and $(\cdot)_{lb}$ for each axis, giving 6 total values for each bounding volume. A safety distance, d is additionally included to account for discrete-time

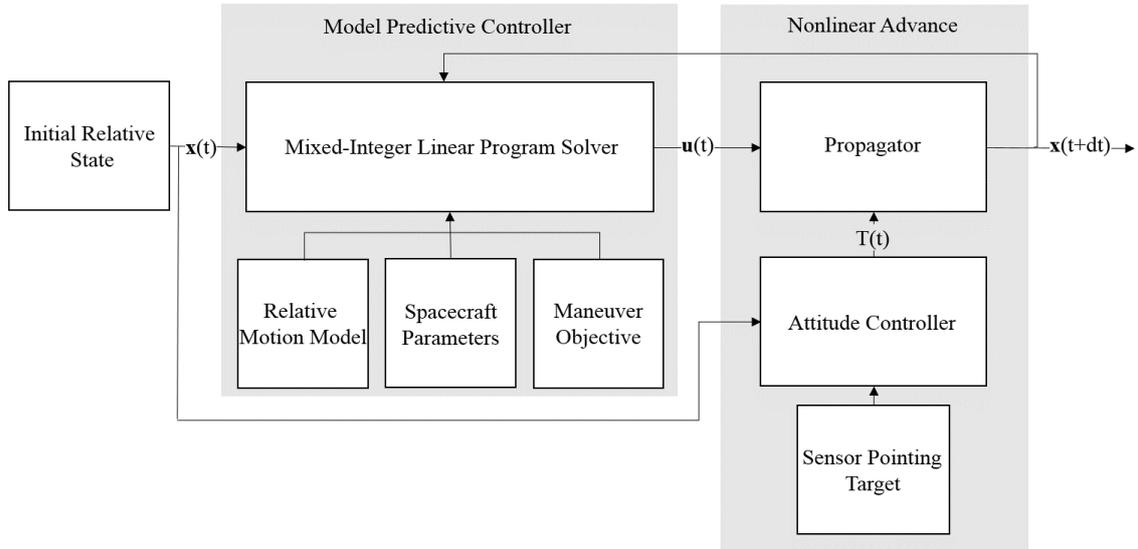


Fig. 2 Diagram of the model predictive control guidance algorithm for obstacle avoidance in short-range proximity operations. $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the controls vector, and $T(t)$ is the time horizon.

trajectory generation. The minimum allowable safety distance, or buffer zone, must be greater than the furthest or longest dimension of the deputy satellite from its own center of mass or greater than $v_{max} dt$ to guarantee the spacecraft and the obstacle bounds never intersect. During calculation, the bounding volume of each obstacle is enlarged in all directions by this buffer.

For each time step within the solver's horizon, the position (x_n, y_n, z_n) must satisfy Eq. (7) to guarantee safe trajectory generation. Equations 7a through 7f are implemented using six binary variables o_k with six inequality constraints where P is an arbitrarily large scalar value, much greater than the scale of the relative position values. This valuation gives the binary constraints the larger impact on the inequalities, of which only 5 must be upheld. Equation 7g is the final inequality that must be defined to constrain the summation of the six binary variables to be less than or equal to

five, ensuring the deputy will not intersect the buffer volume.

$$x_n \leq x_{lb} + P_{o1} \quad (7a)$$

$$y_n \leq y_{lb} + P_{o2} \quad (7b)$$

$$z_n \leq z_{lb} + P_{o3} \quad (7c)$$

$$-x_n \leq -x_{ub} + P_{o4} \quad (7d)$$

$$-y_n \leq -y_{ub} + P_{o5} \quad (7e)$$

$$-z_n \leq -z_{ub} + P_{o6} \quad (7f)$$

$$\sum_{k=1}^6 o_k \leq 5 \quad (7g)$$

E. Attitude Control and Targeted Pointing

Attitude is modeled using quaternions to avoid kinematic singularities. The first three components of the 4×1 quaternion matrix \bar{q} are designated as the Euler axis component q while the fourth component is the scalar component, q_4 . The rate of change of attitude quaternions is defined as Eq. (8) which is a function of the angular velocity of the body, ω , and the 3×3 identity matrix, \mathbf{I}_3 . The rate of change of ω comes from Euler's equations for describing rotational dynamics, written in matrix form in Eq. (9).

$$\dot{\bar{q}} = \frac{1}{2} \begin{bmatrix} q^\times + q_4 \mathbf{I}_3 \\ -q^T \end{bmatrix} \omega \quad (8)$$

$$\dot{\omega} = -\mathbf{I}^{-1} \omega^\times \mathbf{I} \omega + \mathbf{I}^{-1} M \quad (9)$$

where M is the control torque vector to be calculated and \mathbf{I} is the moment of inertia matrix in the b frame. To orient the satellite to the desired pointing target, the desired rotation matrix R_N must be calculated. Once R_N is defined, the corresponding desired quaternions \bar{q}_N can be calculated [7]. A proportional term for the quaternions can be calculated as \bar{q}_p in Eq. (10) using the current attitude

quaternions \bar{q} and \bar{q}_N . [21]

$$\bar{q}_p = \begin{bmatrix} q_{1p} \\ q_{2p} \\ q_{3p} \\ q_{4p} \end{bmatrix} = \begin{bmatrix} q_{4N} & q_{3N} & -q_{2N} & -q_{1N} \\ -q_{3N} & q_{4N} & q_{1N} & q_{2N} \\ q_{2N} & -q_{1N} & q_{4N} & -q_{3N} \\ q_{1N} & q_{2N} & q_{3N} & q_{4N} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (10)$$

A proportional derivative (PD) controller determines the control signals to the CubeSat's attitude control system. A proportional gain k_p and a rate gain k_d are defined. Finally, torque controls M for pointing at the desired coordinate are defined in Eq. 11. [21]

$$M = -k_p \bar{q}_p - k_d \omega \quad (11)$$

III. Results

The deputy satellite was modeled as a 6U CubeSat performing proximity operations around a RSO. The RSO for these simulations is a much larger spacecraft in low-Earth orbit (LEO) with two solar panels, again referred to as the chief. The simulated propellant for the deputy is a monopropellant that is capable of producing 0.25 N of thrust on each face of a CubeSat with a moderate specific impulse, similar to the Aerojet MPS-120. Attitude actuators are simulated with the capability of producing 1 mN-m of torque. A summary of the deputy satellite is included in Table (1). The chief is modeled as a cylindrical body of one meter in radius and two solar panels

Table 1 Summary of 6U CubeSat deputy properties.

Parameter	Dry Mass [kg]	Wet Mass [kg]	Thrust [N]	Isp [s]	Max Torque [mN-m]
Value	6.0	6.5	0.25	150	1.0

of negligible thickness with dimensions of 2x5 meters. The orbit of the chief is circular, with a semi-major axis of 7000 km. The RSO volume bounding the chief is comprised of a 2x2x4 meter volume bounding the body of the craft centered at the origin of the chief's RIC frame. The two solar panels are bounded within two 4x2x1 meter volumes on each side of the centered volume. These

volumes are used as upper and lower bounds of each axis and are given by Eq. 12 below.

$$\begin{aligned}
 x_{lb} &= [-5 \quad -1 \quad 1] \\
 y_{lb} &= [-1 \quad -1 \quad -1] \\
 z_{lb} &= [-0.5 \quad -2 \quad -0.5] \\
 x_{ub} &= [-1 \quad 1 \quad 5] \\
 y_{ub} &= [1 \quad 1 \quad 1] \\
 z_{ub} &= [0.5 \quad 2 \quad 0.5]
 \end{aligned} \tag{12}$$

The first column of each matrix in Eq. 12 corresponds to a solar panel, the second column refers to the spacecraft, and the third column refers to the other solar panel. The safety buffer added to these volumes is $d = 0.5$ meters. For the collision avoidance measures, the arbitrarily large value used is $P = 1 \times 10^6$.

In the first simulation, in §A, the deputy is tasked to perform a relatively simple maneuver of transiting from negative 10 meters along one axis to positive 10 meters along the same axis without a chief to avoid. This 20 meter maneuver is repeated for each axis, and serves as an example of the finite waypoint maneuvering abilities without collision avoidance or attitude pointing.

In the following simulations, in §B, the deputy is tasked to maneuver from an initial position of 10 meters behind the chief to 10 meters ahead, in-track, of the chief using collision avoidance. This example is simulated first with no attitude pointing requirements (§B1). Next, the deputy is commanded to direct its positive \hat{b}_1 axis towards the center of the chief while it completes the same waypoint maneuver (§B2). Then, to demonstrate reaction to new information, the deputy repeats the same maneuver with attitude pointing, however, new information will be presented to the deputy after a designated time to represent a sudden rotation in the chief's solar panels (§B3). Finally, in §C, a multi-waypoint maneuver is simulated as a comprehensive test of safe trajectory planning, targeted attitude pointing, and reaction to new information while performing a detailed circumnavigation of the chief.

All simulations were run in real-time using Matlab 2015a on a laptop operating on battery power with an Intel(R) Core(TM) i5-4210U CPU processor at 1.760GHz.

A. Waypoint Maneuver

A finite maneuver was analyzed without collision avoidance or attitude pointing to demonstrate the finite waypoint maneuvering abilities of the algorithm. The deputy was tasked to travel 20 meters along each axis from negative 10 meters along the axis to positive 10 meters along the same axis. Figure 3 displays a comparison of this maneuver for each direction. The figure shows the total maneuver Δv vs the total time-of-flight for a point-to-point maneuver consisting of the 20 meter transit. For the radial maneuver in the x-axis, the solver found the optimal solution starting at half

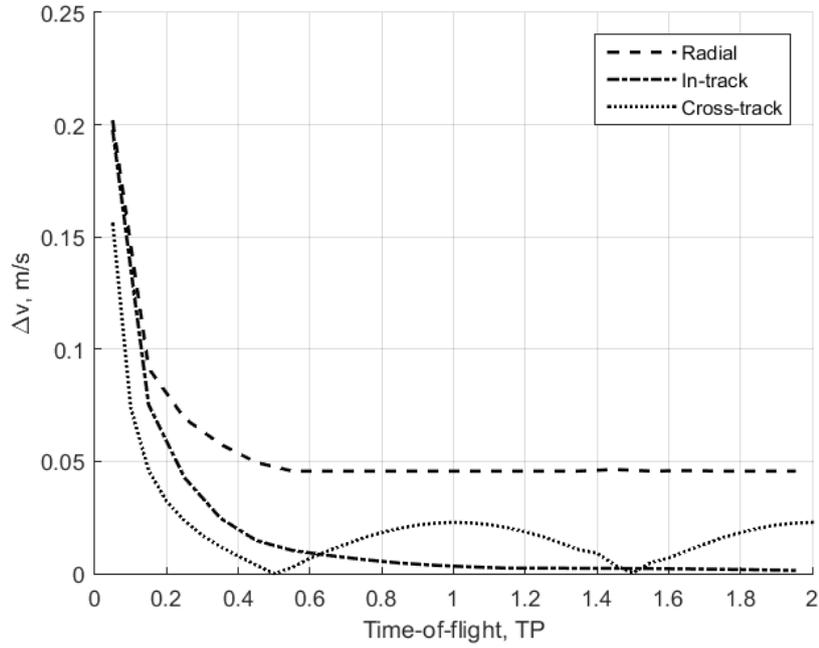


Fig. 3 Total Δv vs Time-of-Flight for a 20 meter finite waypoint maneuver along three axes without accounting for collision avoidance. The time period (TP) for this simulation was $TP = 5.504403 \times 10^3 s$.

an orbital period for a total Δv equal to $2 \times 2nx_0$, corresponding to the velocity necessary to begin and exit a relative motion ellipse. The radial direction has the highest Δv requirement of all the axes. This is due to the fact the radial direction points directly out behind the deputy from the origin, and therefore falls furthest from the orbital direction, or the direction in which the deputy's orbit naturally moves. Because of this, it takes the most fuel and requires the most Δv to maneuver in the radial direction.

During the in-track maneuver in the y-axis, the Δv required approaches zero as the maneuver is given more time. This is due to the fact the in-track direction is oriented most closely to the orbital direction (see Figure 1), and therefore the deputy only requires small adjustments to maneuver in this direction as it continues its orbit.

The cross-track maneuver in the z-axis successfully solves the optimal transfer of zero Δv required for odd multiples of $n(TP/2)$ where the satellite will drift to the desired waypoint. This is due to the orientation of the cross-track direction (again, see Figure 1). If the deputy moved solely in the cross-track direction, it would 'cut the corner' of its own orbit to meet up again with it at a later time. This timing happens in odd multiple of $n(TP/2)$ so therefore the deputy needs zero fuel, and therefore zero Δv , at these points.

B. Waypoint Maneuver with Collision Avoidance

In the following simulations the deputy is tasked to maneuver from an initial position of 10 meters behind the chief to 10 meters ahead, in-track, of the chief using collision avoidance.

1. Without Attitude Pointing

For this scenario, the chief's solar panels are positioned in the y-z plane. Figure 4 illustrates the deputy's trajectory relative to the chief. The entire maneuver was completed in 93 seconds and consumed 7.5 grams of fuel. For comparison, the computational time it took to calculate this trajectory was 20 seconds, which is about 4.5 times faster than real-time. The control signals and attitude signals of this trajectory are seen in Figures 5 and 6, respectively. The pointing, or pose, of the deputy is plotted in five second intervals. The deputy accelerates to the maximum allowed relative velocity from the zero-relative velocity initial state during the first 12 seconds of the maneuver. The maneuvering craft then performs a series of burns, starting at about 15 seconds in, to avoid collision with the main cylindrical body of the chief. The majority of these burns occur in the \hat{b}_1 and \hat{b}_2 directions, as the in-track (\hat{b}_1) direction is that which the deputy is tasked to maneuver in, while the cross-track (\hat{b}_2) direction is that which the deputy chooses to avoid the chief in. The radial direction (\hat{b}_3) sees the minority of controls due to the results seen from §A in which no collision avoidance was used. The deputy is optimizing for fuel usage, or minimizing Δv , and

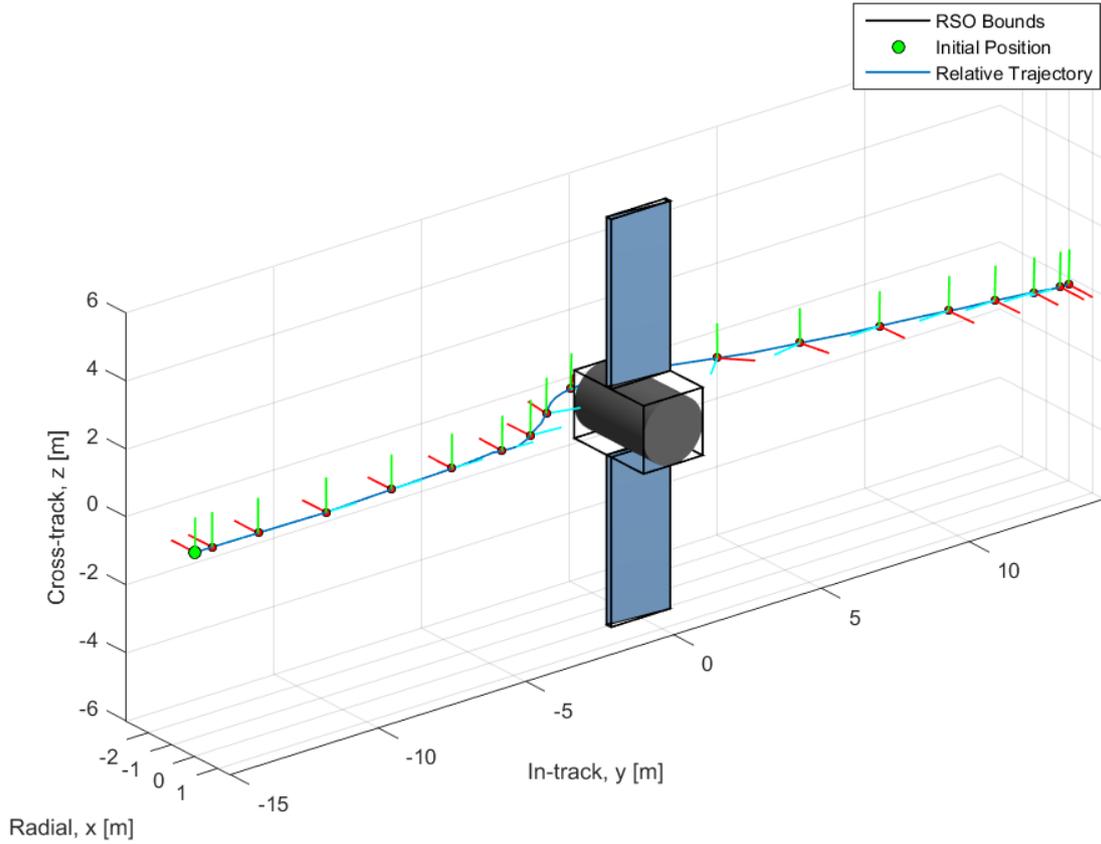


Fig. 4 Relative trajectory of deputy satellite exhibiting obstacle avoidance with no targeted attitude pointing while performing a waypoint-to-waypoint maneuver around the chief. The chief is bounded within the RSO volume bounds produced by the collision avoidance algorithm. The position of the deputy is plotted in five second intervals. The deputy's axes are colored as follows: \hat{b}_1 is blue, \hat{b}_2 is red, and \hat{b}_3 is green.

the radial direction consumes the most in terms of maneuverung. Therefore the deputy chooses to maneuver in the cross-track direction before successfully completing its maneuver in the in-track direction.

2. With Attitude Pointing

The same waypoint maneuver was simulated, this time with the deputy commanded to orient its \hat{b}_1 axis to the center of the chief. Since the chief is centered at the local RIC origin, the deputy simply attempts to target the origin. The \hat{b}_2 axis of the deputy is constrained to remain in the x-z

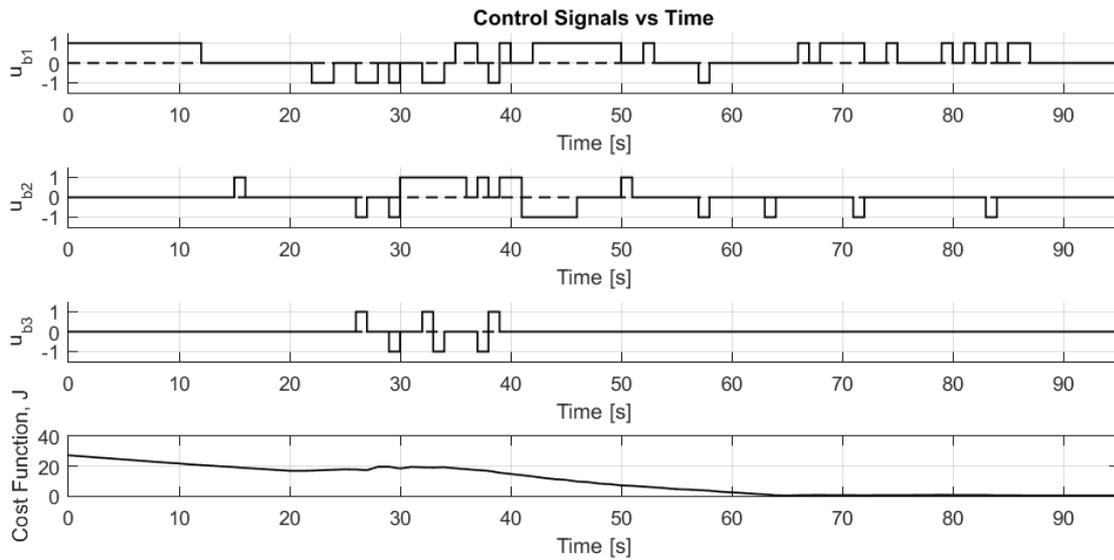


Fig. 5 Control signals of the deputy satellite and cost function of the MPC system for the maneuver in Figure 4.

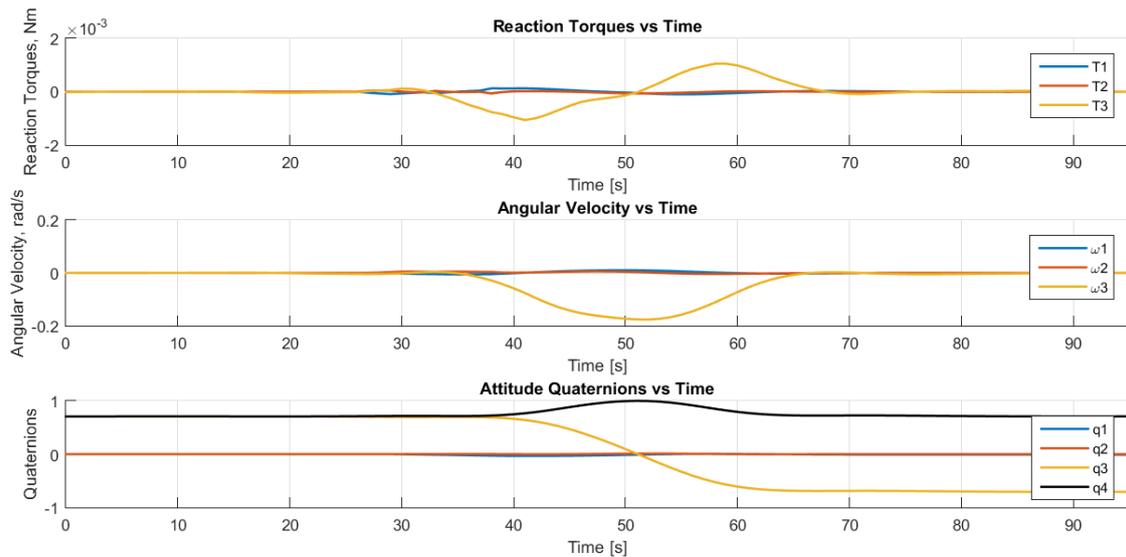


Fig. 6 Attitude signals of the deputy satellite for the maneuver in Figure 4.

plane. This trajectory with the targeted attitude command is pictured in Figure 7. As before, the pose of the deputy is plotted every five seconds. The maneuver was completed in a total time of 75 seconds and 12.4 grams of fuel was consumed, or roughly 165% of the fuel consumed without attitude pointing. Again for comparison, the computational time it took to calculate this trajectory

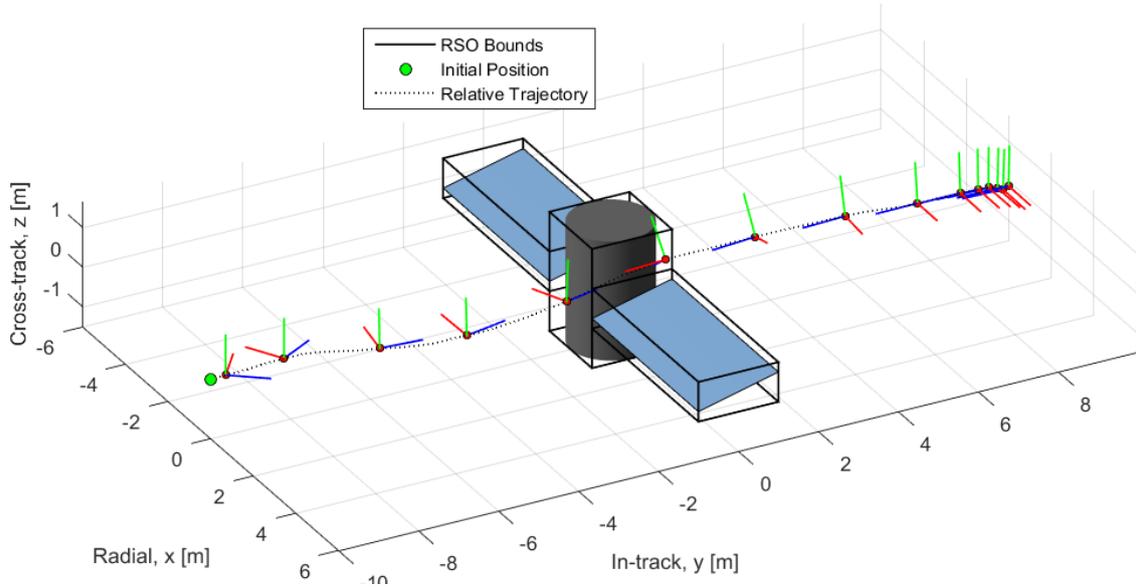


Fig. 7 Relative trajectory of deputy satellite exhibiting obstacle avoidance with targeted attitude pointing while performing a waypoint-to-waypoint maneuver around the chief. The chief is bounded within the RSO volume bounds produced by the collision avoidance algorithm. The position of the deputy is plotted in five second intervals. The deputy's axes are colored as follows: \hat{b}_1 is blue, \hat{b}_2 is red, and \hat{b}_3 is green.

was 17 seconds, again about 4.5 times faster than real-time. In this maneuver, the deputy again attempts to avoid the chief with a thrust into the cross-track direction instead of the radial to optimize its fuel usage. However this time, the deputy must maintain its \hat{b}_1 axis attitude pointing to the center of the chief, so it must include additional controls to rotate and produce a torque in its radial axis, or \hat{b}_3 direction. This is the cause for the additional fuel consumption, and takes about 65% more Δv than the maneuver did without any attitude pointing. Again the majority of controls are spent in the in-track and cross-track axes.

3. *With Attitude Pointing in Partially Unknown Environment*

The previous scenario is repeated, however after 20 seconds of simulation time, the chief's solar panels instantaneously rotate 90° from the x-y plane to the x-z plane. Figure 8 shows the new trajectory updated during the maneuver. The deputy's knowledge of chief's solar panels is instantaneously updated, alerting the deputy that the panels now lie in the x-z plane. Prior to the

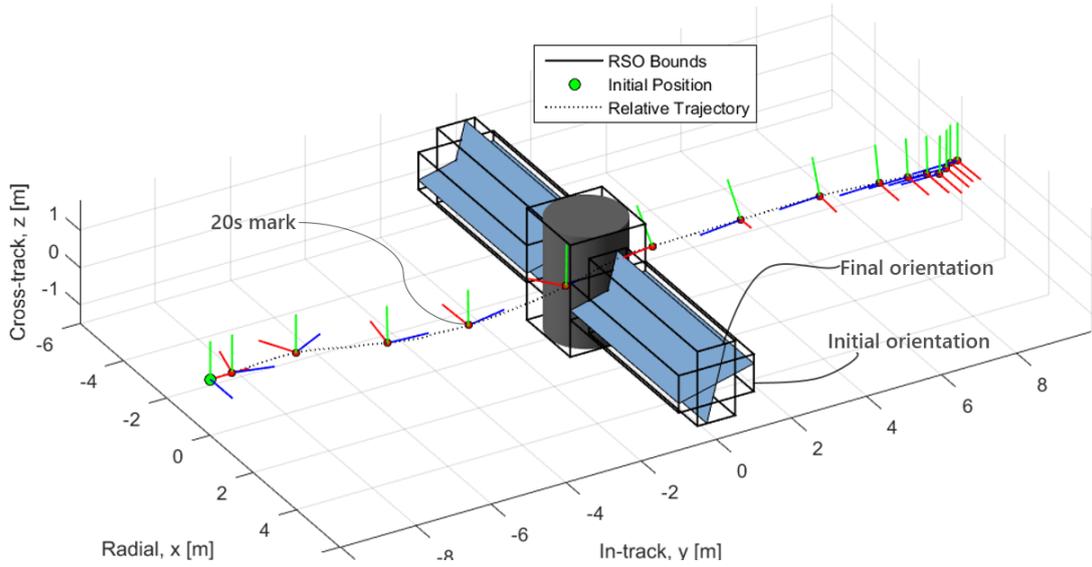


Fig. 8 Relative trajectory of deputy satellite exhibiting obstacle avoidance with targeted attitude pointing while performing a waypoint-to-waypoint maneuver around the chief. The chief is bounded within the RSO volume bounds produced by the collision avoidance algorithm. The deputy reacts to the solar panels rotating 20 seconds into the simulation. The deputy’s position is plotted in five second intervals. The deputy’s axes are colored as follows: \hat{b}_1 is blue, \hat{b}_2 is red, and \hat{b}_3 is green.

20 second mark, the deputy exhibits the same behavior as the previous scenario in Figure 7. The satellite has planned to avoid the chief and its solar panels by making the same small maneuver thrusting into the cross-track direction and around the right solar panel to reach -10 meters in-track. However, after the 20 second mark, the chief’s solar panels have rotated into the cross-track plane thereby blocking the deputy’s previously planned route. At this time the deputy re-examines the trajectory and begins a sudden deceleration and rerouting to avoid collision. The maneuver was completed in a total time of 79 seconds and 12.6 grams of fuel was consumed. This is only slightly higher than the prior situation, consuming just 2% more fuel than without the sudden solar panel rotation, and about 68% more than without attitude pointing. The computational time for this trajectory was 18 seconds, also about 4.5 times faster than real-time.

C. Multi-Waypoint Maneuver

To demonstrate multi-waypoint maneuvering abilities, the deputy was tasked to enter a series of relative motion ellipses around the chief. This mission uses both the finite maneuver solver as well as the model predictive control proximity operations algorithm to illustrate appropriate uses for each guidance method and the abilities of CubeSats (which the deputy is modeled as) to dock with significantly larger spacecraft. An example of these capabilities is demonstrated in Figure 9. The deputy begins with zero relative velocity with offsets in each axis of 200 meters radial, -100

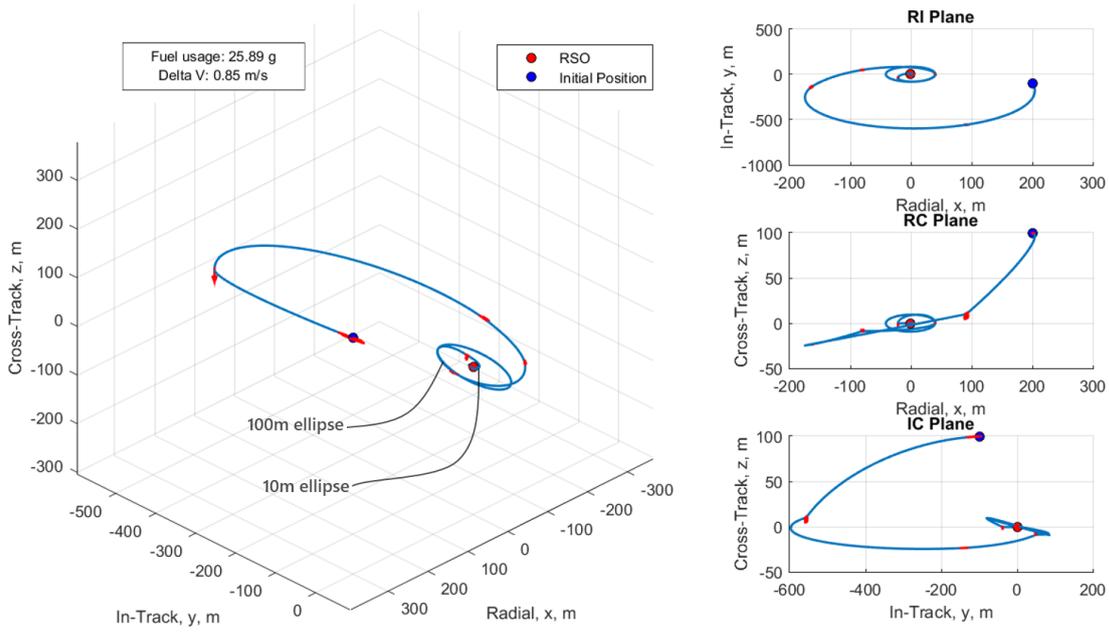


Fig. 9 Multi-waypoint, or docking, maneuver to approach the chief, a larger RSO. The deputy enters a series of relative-motion ellipses before beginning a final docking approach in the anti-radial direction, $-\{\hat{e}_r\}$.

meters in-track, and 100 meters cross-track. The deputy then uses the open-loop guidance method to insert itself into a relative motion ellipse with a semi-major axis of 100 meters. The maneuver is given a defined time of exactly one orbital period. The deputy can remain in this ellipse with very little station keeping fuel required to observe the chief and recharge its batteries. Next, the same guidance method is used to command the deputy from the 100 meter ellipse into an ellipse with a semi-major axis of 10 meters. Again, the deputy can remain here to perform operational checks and

recharge. Finally, model predictive control is used to perform the final docking approach and guide the deputy safely towards the $-\hat{e}_r$ face of the chief. The complete maneuver consumed 25.9 grams of fuel and included a total Δv of 0.85 meters per second. This is about two times or 200% of the fuel consumption from the maneuver in Figure 8. However there were three maneuvers performed in the complete docking simulation versus only one in the prior, showcasing the efficient ability of CubeSats for RPO missions.

IV. Conclusions

Cost functions and constraints for both open and closed-loop rendezvous and proximity operation behaviors were presented using mixed-integer linear programming. An overview of the collision avoidance and attitude control implemented in the model were additionally covered.

A demonstration of the open-loop discrete-time trajectory algorithm was presented to demonstrate fuel-efficient use of relative motion to perform point-to-point maneuvers around an orbiting reference frame. Next, the model predictive control algorithm demonstrated capabilities of generating collision-safe trajectories while maintaining attitude pointing at a reference satellite. The MPC algorithm additionally was simulated reacting to new information in a partially unknown environment. Finally, a comprehensive rendezvous and docking operation was simulated to illustrate appropriate uses for each method of guidance and the abilities of CubeSats to dock with significantly larger spacecraft.

At a time step of one second, the model can achieve low-fidelity controls. The model is valid for eccentric and near-circular orbits, however, perturbations from third bodies and from J2 are not considered. The bounding collision avoidance volumes must be defined from an external information source, with real-time updates when these bounding volumes must be changed in accordance with movement from the chief spacecraft.

The work described above provides an innovative and highly accurate solution to the question of safe-trajectory generation for satellite proximity operations. The integration of attitude pointing provides a high fidelity analysis of the controller's ability to conduct proximity operations. This work gives an accurate model for orbital and attitude dynamics and control. Additionally, the linear

set of orbital equations used in the above model provides a realistic orbit determination when linear approximations are appropriate. The equations allows the model to take small eccentricities into account, producing a reliable orbit prediction and fuel loss calculation. Overall, the closed form MILP combined with MPC shows its potential to outperform other optimal control methods in terms of predictability, computational effort, and simplicity. Future work may increase the fidelity of the simulation, including modeling the motion of both spacecraft with a more robust set of equations, such as the NERM, by performing a Hardware In the Loop (HWIL) simulation, or by using sophisticated sensing and estimation techniques. The model may also be improved by incorporating the Gim-Alfriend State Transition Matrix to account for perturbations from J2. Additionally, the attitude controller can be re-assigned from a PD controller to a model predictive controller. This change will allow the attitude controller to use more information from the translational trajectory solver and allow for more control over attitude pointing.

Ephemeris files can be generated automatically in Matlab and directly exported to Systems Tool Kit (STK) for further analysis. STK can provide thermal, power generation, and radiation information about the maneuver generated by the algorithm in Matlab.

Acknowledgments

This material is based upon work supported by the Virginia Tech College of Engineering New Horizons Graduate Scholars Program, the Walts Graduate Fellowship, the Engineering Mechanics Program, and the Kevin T. Crofton Aerospace and Ocean Engineering Department. The authors would like to express their thanks to the Virginia Tech Ross Dynamics Lab and Center for Space Science and Engineering Research members for their helpful discussions and edits.

References

- [1] Zhu, Y., *Efficient Nonlinear Optimization with Rigorous Models for Large Scale Industrial Chemical Processes*, Vol. 72, 2011.
- [2] Qin, S. J. and Badgwell, T. A., "A survey of industrial model predictive control technology," *Control engineering practice*, Vol. 11, No. 7, 2003, pp. 733–764.

- [3] Hutter, F., Hoos, H. H., and Leyton-Brown, K., “Automated configuration of mixed integer programming solvers,” in “International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming,” Springer, 2010, pp. 186–202.
- [4] Richards, A., Schouwenaars, T., How, J. P., and Feron, E., “Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 755–764.
- [5] Koon, W. S., Lo, M. W., Marsden, J. E., and Ross, S. D., *Dynamical Systems, the Three-Body Problem and Space Mission Design*, Marsden Books, 1st ed., 2011.
- [6] Schaub, H. and Alfriend, K., “Impulsive feedback control to establish specific mean orbit elements of spacecraft formations,” *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 4, 2001, pp. 739–745.
- [7] Schaub, H. and Junkins, J., *Analytical Mechanics of Space Systems*, AIAA, Reston, Virginia, 2nd ed., 2009.
- [8] Yan, H. and Gong, Q., “Feedback Control for Formation Flying Maintenance Using State Transition Matrix,” *The Journal of the Astronautical Sciences*, Vol. 59, No. 1-2, 2012, pp. 177–192.
- [9] Tillerson, M., Inalhan, G., and How, J., “Co-ordination and Control of Distributed Spacecraft Systems Using Convex Optimization Techniques,” *International Journal of Robust and Nonlinear Control*, Vol. 12, No. 2-3, 2002, pp. 207–242.
- [10] Mueller, J. B., Griesemer, P. R., and Thomas, S. J., “Avoidance maneuver planning incorporating station-keeping constraints and automatic relaxation,” *Journal of Aerospace Information Systems*, Vol. 10, No. 6, 2013, pp. 306–322.
- [11] Schouwenaars, T., *Safe Trajectory Planning of Autonomous Vehicles*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2006.
- [12] Rogers, A., *Optimization-Based Guidance For Satellite Relative Motion*, Ph.D. thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2016.
- [13] Nastasi, K. M., Thomas, D., Tetreault, K., Elliott, I., and Black, J., “Real-Time Optimal Control, & Tracking of Autonomous Micro-Satellite Proximity Operations,” in “AIAA SPACE 2016,” p. 5617, 2016.
- [14] Thomas, D. J., Tetreault, K., Mott, K., Elliott, I., Scheible, R., Ohriner, E., Nastasi, K. M., and Black, J. T., “Real-Time On-board Estimation & Optimal Control of Autonomous Micro-Satellite Proximity Operations,” in “55th AIAA Aerospace Sciences Meeting,” , 2017, p. 0398.
- [15] Roscoe, C., Westphal, J., Shelton, C., and Bowen, J., “CubeSat Proximity Operations Demonstration (CPOD) Mission: End-to-end Integration and Mission Simulation Testing,” in “AIAA Astrodynamics Specialist Conference, August 2015,” , 2015.

- [16] Gim, D.-W. and Alfriend, K. T., "State transition matrix of relative motion for the perturbed noncircular reference orbit," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, 2003, pp. 956–971.
- [17] AeroJet, "MPS-120XW CubeSat High-Impulse Adaptable," , 2016.
<https://www.rocket.com/cubesat/mps-120xw>.
- [18] VACCO, "Standard Micro Propulsion System: Standard MiPS," , 2016.
<http://www.vacco.com/space/chems-overview>.
- [19] Vallado, D., *Fundamentals of Astrodynamics and Applications*, Space Technology Library, Microcosm Press, Hawthorne, California, 4th ed., 2013.
- [20] Ghosh, P., "Spacecraft Flying in Formation," , 2016. Wolfram Demonstration Project.
- [21] Blanke, M. and Larsen, M. B., "Satellite dynamics and control in a quaternion formulation," Tech. rep., Technical University of Denmark, Department of Electrical Engineering, 2010.