SPACE Conferences and Exposition
13 - 16 September 2016, Long Beach, California
AIAA/AAS Astrodynamics Specialist Conference

Trajectory Optimization for Proximity Operations Around Tumbling Geometrical Constraints via Legendre Polynomials

Jian-Feng Shi,* Steve Ulrich†
Carleton University, Ottawa, Ontario K1S 5B6, Canada
Gregory E. Chamitoff,‡ Benjamin J. Morrell§
University of Sydney, Sydney, New South Wales 2006, Australia
Andrew Allen¶
MacDonald, Dettwiler and Associates Ltd., Brampton, Ontario, L6S 4J3, Canada

This paper presents the development of an optimization guidance algorithm capable of generating three-dimensional trajectories enabling a small free-flyer robotic spacecraft to maneuver in close proximity to translating and tumbling satellites. Specifically, the proximity guidance law parameterizes the velocity trajectory using Legendre polynomials and optimizes their coefficients in a two-step fashion. In the first step, a sub-optimal solution that satisfies the boundary conditions, dynamics constraints, and performance limitations is obtained. In the second step, time-permitting, the optimal solution which also minimizes the path length is obtained. The performance of the guidance law is evaluated in simulation results for a fly-around scenario of both a non-cooperative stationary and tumbling target satellite.

I. Introduction

Spacecraft robotics systems with the capability to safely navigate around dynamic obstacles using real-time decision-making algorithms without human support and intervention represent key techologies to prepare for or to support the capture of large, geometrically-complex objects such as Envisat for active orbital debris mitigation. Furthermore, due to advancements in electronic miniaturization, such challenging missions may utilize multiple nanosatellite-sized inspectors deployed on-orbit by a single larger, host servicer spacecraft.[1] As an example of an enabling capability, the US Defense Advanced Research Project Agency (DARPA) is currently partnered with MDA Robotics and Automations in investigating a Payload Orbital Delivery (POD) system.[a] The POD system provides an on-orbit payload deployment and release function using a novel debris-free release and guide mechanism that offers user-selectable separation velocity while inhibiting undesired tip off and tumbling independent of internal payload mass and offset. Such delivery systems could be enhanced to support an intelligent nano-satellite inspector. The autonomous inspector spacecraft may work cooperatively to carry out inspection or maintenance tasks near a high value asset with complex appendages and geometry (e.g., geostationary satellites or space stations) while minimizing the risks of accidental collisions. In addition, in cases of large-scale space structure maintenance, the use of inspector spacecraft could eliminate the need for various on-board cameras while minimizing fuel usage for the host

*Ph.D. Candidate, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Member AIAA.
†Assistant Professor, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Senior Member AIAA.
‡Lawrence Hargrave Professor, School of Aerospace, Mechanical and Mechatronic Engineering. Professor of Practice, Aerospace Engineering, Texas A&M University. Associate Fellow AIAA.
§Ph.D. Candidate, School of Aerospace, Mechanical and Mechatronic Engineering. Student Member AIAA.
¶Manager, Guidance, Navigation, and Control Department, 9445 Airport Road.
[a]Phoenix Release POD Interface Requirements as First Step Toward Vision of FEDEX TO GEO, http://www.darpa.mil/news-events/2014-11-10 [retrieved 25 January 2016]

servicing spacecraft. In such inspection missions, an on-board guidance algorithm would be highly valuable to enable the inspector spacecraft to autonomously maneuver in close proximity to tumbling satellites. To address this specific problem, previous work proposed various nonlinear constrained optimization techniques. For example, a thrust limited maneuver with path constraint approach was developed by Taur et al.,[2] an adaptive artificial potential function-based strategy was developed by Munoz and Fitz-Coy,[3] an example of an optimization technique for spacecraft proximity maneuvers with interior point constraints was presented by Ulybyshev,[4] a minimum relative distance constraint via indirect optimization was employed by Hadaegh et al.,[5] a spline trajectory was used by Hadaegh and Singh,[6] the so-called mixed-integer linear programming (MILP) was developed by Richard et al.,[7] and Breger and How,[8] a genetic algorithm was investigated by Luo et al.,[9,10] a sparse optimal control software (SOCS) was validated by Boeing,[11] and second-order cone programming (SOCP) techniques by Alizadeh et al.,[12] Boyd et al.,[13] and Lu and Liu.[14]

Another recent solution to the nonlinear constrained optimization problem is a guidance law termed Admissible Subspace Trajectory Optimizer (ASTRO), which consists of an optimal / sub-optimal collision avoidance path-planning strategy that also considers spacecraft performance restrictions.[15] ASTRO transforms the problem into a parameter space that is well behaved; it can navigate complex surroundings with multiple obstacles and constraints. This ASTRO algorithm was implemented and tested on the International Space Station (ISS) Synchronized Position Hold Engage Re-orient Experimental Satellites (SPHERES). SPHERES is a six degrees of freedom autonomous spacecraft testbed that operates in the ISS microgravity envioronment.[16] The ASTRO SPHERES experiment demonstrated the feasibility to solve in real-time a path-planning problem for collision avoidance and spacecraft performance restriction using optimal or near optimal solutions. During the development of the ASTRO algorithm, the multi-dimensional optimization problem for spacecraft proximity operations was studied. An interesting feature of the ASTRO algorithm is the ability to provide constraints on the interior of the path rather than simply terminal constraints. The original ASTRO algorithm was experimentally validated through the ISS-SPHERES testbed, by executing a compiled version of the algorithm in MATLAB while commanding the SPHERES satellite via wireless command link at one second intervals.[15] The computation of the trajectory could however take up to ten seconds to complete. The original development considered a spherical object traveling with varying velocities within cylindrical corridors in close proximity to elementary spherical or cylindrical obstacles.[17,18] As such, the geometrical constraints did not provide geometric pose motion restrictions. In this context, this paper bridges the gap between the use of constraints purely on the translation of the spacecraft and the use of more complex and realistic geometrical constraints that are representative of tumbling uncooperative satellites. Additionally, improvements are proposed to the algorithm to maximize its efficiency in computational speed, so it is better-suited for real-time operations.

This paper is organized as follows: Section II defines the autonomous trajectory guidance problem. Section III summarizes the theory upon which the optimal guidance law is developed. Then, Sec. IV describes the modifications proposed herein to accommodate complex, rotating geometrical constraints. Section IV describes the MATLAB simulation environment and provides an overview of the software program architecture. Section V provides simulation results demonstrating the performance of the modified ASTRO algorithm for an inspector spacecraft navigating around stationary and tumbling satellites.

II. Problem Statement

The objective of the guidance algorithm is to compute a feasible trajectory to be tracked by a small inspector spacecraft operating in close proximity to a large and complex target satellite that is both translating and rotating in space. Geometric constraints, such as those imposed by the satellite's solar arrays sweeping out a region of space, should be avoided by the inspector spacecraft. On the other hand, the path computed by the guidance law should minimize distance between the inspector spacecraft and the target satellite for close inspection. Since the modification to the ASTRO algorithm presented here is focused on the path planning aspect of trajectory guidance, it is assumed the inspector spacecraft has already performed a visual inspection of the target satellite for a sufficiently long time and, as such, has the ability to estimate the mass properties of the target satellite and propagate its dynamics forward. This can be done, for example, through SLAM- or iSAM-based techniques as demonstrated through the SPHERES-VERTIGO experiment on ISS.[19] The propagation time is bounded by the initial guess of overall translational maneuver time of the inspector spacecraft as it travels from its initial to final position.

The original ASTRO algorithm solves a translational guidance problem in two iterative steps. The first

step calculates a sub-optimal, feasible, trajectory that simultaneously: (1) satisfies boundary conditions, (2) avoids geometrical constraints, and (3) meets spacecraft-imposed performance limitations such as maximum thruster accelerations. The second step attempts to achieve an optimal trajectory that also minimizes the path length.

The proposed modification to the original ASTRO algorithm provides a means to determine the appropriate geometric constraint for ASTRO solver planning. During the initialization phase, the algorithm evaluates the geometric body of the target satellite and its motion, to form the geometric constraints. Specifically, the modified guidance law algorithm can be broken down into three steps:

1. the prediction and propagation of the target satellite attitude and translational motion;
2. the parameterization of the trajectory path using Legendre polynomials, as well as modification of the path based on inner spatial and/or performance constraints; and
3. the satisfaction of boundary conditions and optimization based on projected gradient search.

III. Admissible Subspace Trajectory Optimizer Theory

The general theory behind the ASTRO guidance algorithm is summarized here for completeness. For a more detailed derivation of the theory, the reader is referred to the work of Chamitoff et al.[15] The key feature of the ASTRO algorithm is its ability to optimize a trajectory-related cost function by projecting its gradient onto the subspace of parametric variation that enforces the boundary conditions. Hence, subsequent iterations move closer to a solution that satisfies all constraints. Weights are used in the cost function so constraint violation gradients dominate the parameter search. Additionally, by taking advantage of special properties of Legendre polynomials, the path-length penalty functions can ensure a positive-definite cost function with respect to the higher dimensional space of parameter errors, which, in most cases, guarantees asymptotic convergence directly to an admissible and optimal solution. Furthermore, the Legendre polynomials transform the path optimization problem into a parameter optimization problem that uses the Legendre orthogonality properties to greatly simplify the cost function, allowing rapid computation.

A. Augmented Cost Function

The boundary conditions are expressed as

$$\begin{aligned} f_{BC_1}(\mathbf{x}_0(t_0), \dot{\mathbf{x}}_0(t_0)) &= 0 \\ f_{BC_2}(\mathbf{x}_f(t_f), \dot{\mathbf{x}}_f(t_f)) &= 0 \end{aligned} \tag{1}$$

where $\mathbf{x}_0(t_0)$ and $\dot{\mathbf{x}}_0(t_0)$ are the initial position and velocity of the inspector spacecraft, respectively, and $\mathbf{x}_f(t_f)$ and $\dot{\mathbf{x}}_f(t_f)$ are the final position and velocity. The geometric, velocity, and acceleration constraints are described by

$$f_{c_j}(\mathbf{x}(t), \dot{\mathbf{x}}(t), \ddot{\mathbf{x}}(t)) \leq 0, \quad \forall j \in [1, n], \quad \forall t \in [t_0, t_f] \tag{2}$$

where n is the number of constraint functions. Morrell et al.[17] provide an example of how a local dynamic constraint field can be formulated. The inner loop of the algorithm satisfies Eq. (1) and Eq. (2), while the outer loop optimizes the solution by minimizing the augmented cost function given by

$$J = f_s^2(S) + \sum_{j=1}^{n} K_j \max_{t\in[t_0,t_f]} f_{c_j}^2 \tag{3}$$

where $f_s^2(S)$ is the path length cost function that minimizes the path length, which is given by

$$S^2 = \int_{t_0}^{t_f} \sum_{i=1}^{3} \dot{x}_i(t)^2 \mathrm{d}t \quad \forall i \in [1, 2, 3] \tag{4}$$

where i is the axis direction, and the coefficients K_j are the relative weights W_j for each constraint function, and are given by

$$K_j = \begin{cases} 0, & \text{if } f_{c_j} \leq 0 \\ W_j, & \text{if } f_{c_j} > 0 \end{cases} \tag{5}$$

The weights are selected to ensure the constraints dominate the cost function. When the second term in Eq. (3) is zero, then the solution is admissible, regardless of the path length cost function value.

B. Legendre Polynomial Parameterization

Equation (3) can be simplified by parameterizing the trajectory as Legendre polynomials and normalizing the time interval. Taking advantage of the Legendre polynomial property

$$\int_{-1}^{1} P_m(x)P_n(x) = \begin{cases} 0, & m \neq n \\ \frac{2}{2n+1}, & m = n \end{cases} \tag{6}$$

the continuous velocity function $\dot{x}_i(t)$ can be parameterized as

$$\dot{x}_i(t') = \sum_{k=0}^{N} C_{ik} P_k(t') \tag{7}$$

C_{ik} is the k^{th} coefficient of the Legendre polynomials of order N that defines the inspector spacecraft trajectory along the i^{th} axis. The normalized time is denoted as t' and is defined as

$$t' = 2\left[\frac{t - t_0}{t_f - t_0}\right] - 1 \tag{8}$$

The upper bound path length is written as

$$\bar{S}^2 = \sum_{i=1}^{3}\sum_{k=0}^{N}\left\{C_{ik}^2\left(\int_{-1}^{1}[P_k'(t')]\,\mathrm{d}t'\right)^2\right\} \tag{9}$$

Since P_k' are the standard Legendre polynomials, the integral can be evaluated off-line. Using the orthogonality of the Legendre function as described in Eq. (6), all cross terms are zero which reduces the computation by a factor of N. Finally, the function in Eq. (3) is reformulated as

$$J = \sum_{j=1}^{n+1}[f_j(\mathbf{C})]^2 \tag{10}$$

where the rows of $\mathbf{C} = \begin{bmatrix} C_{11} & \cdots & C_{1N} \\ \vdots & \ddots & \vdots \\ C_{31} & \cdots & C_{3N} \end{bmatrix}$ correspond to coefficients of the Legendre polynomials, and f_j represents the maximum violation for each constraint function, with the $n+1$ term being the path length cost term. Let C_{ik}^* be optimal values of the coefficients for the optimal trajectory. Then, the optimal cost solution is

$$J^* = \sum_{j=1}^{n+1}[f_j(\mathbf{C}^*)]^2 \tag{11}$$

and $J' = J^* - J$ is positive-definite with respect to the coefficient errors $\delta C_{ik} = C_{ik} - C_{ik}^*$ if

$$\frac{\partial J'}{\partial C_{ik}} = 2\sum_{j=1}^{n+1} f_j(C_{ik})\frac{\partial f_j}{\partial C_{ik}} \text{ is monotonic} \tag{12}$$

or, equivalently

$$\frac{\partial^2 J'}{\partial C_{ik}^2} = 2\sum_{j=1}^{n+1}\left(\frac{\partial f_j}{\partial C_{ik}}\right)^2 + f_j(C_{ik})\frac{\partial^2 f_j}{\partial C_{ik}^2} \geq 0 \tag{13}$$

Then, C_{ik}^* is a unique minimum for J', that is, for any $C_{ik} \neq C_{ik}^*$, J' can be reduced by a discrete step in C_{ik}, where $[C_{ik}]_{new} = [C_{ik}]_{old} + \delta C_{ik}$, such that

$$\delta C_{ik} = -\alpha\left[\frac{\partial J'}{\partial C_{ik}}\right] + \beta_{ik} \tag{14}$$

where $\alpha > 0$ and $\beta_{ik} \neq 0$. Finally, the pseudo-gradient search in Eq. (14) will converge to $\delta C_{ik} = 0$. It shall be noted the optimization of J' effectively optimizes J. Furthermore, it can be shown that any constraint $g(\mathbf{X})$ that is a convex function of the states, derivatives, and/or accelerations is allowable to maintain convexity, and that if the matrix $\partial^2 g/\partial \mathbf{X}^2$ is strictly positive definite, then $(\partial^2 J')/(\partial C_{ik}^2)$ will also be strictly positive definite, and a gradient-based search would encounter $\partial J'/\partial C_{ik} = 0$ only at $\delta C_{ik} = 0$. It is possible, however, for some constraint geometries to violate the convexity conditions. Nevertheless, with the transformation to the higher dimensional space, the ASTRO algorithm is usually still able to converge to the optimal solution.[18]

C. Boundary Conditions and Projected Gradient Search

To generate admissible sub-optimal solutions as quickly as possible, a gradient projection is used to ensure that the boundary conditions at t_0 and t_f are always met. From the trajectory parameterization of Eq. (7), the boundary conditions can be written as

$$\mathbf{X}_{BC} = \mathbf{P}_{BC}\mathbf{C} \tag{15}$$

or

$$\mathbf{X}_{BC} = \mathbf{P}_{BC}\mathbf{C}_{N\times 3} = \mathbf{P}_{BC}\left\{\mathbf{C}_{\perp_{N\times 3}} + \mathbf{C}_{||_{N\times 3}}\right\} \tag{16}$$

where $\mathbf{C}_\perp$ are in the null-space of $\mathbf{P}_{BC}$, such that

$$\mathbf{P}_{BC}\mathbf{C}_{\perp_{N\times 3}} = 0 \tag{17}$$

and any variations of $\mathbf{C}_\perp$ will not disturb the boundary conditions. Finally, the projected gradient step is

$$\delta\mathbf{C}_\perp = \left[\mathbf{I} - \mathbf{P}_{BC}^T\left(\mathbf{P}_{BC}\mathbf{P}_{BC}^T\right)^{-1}\mathbf{P}_{BC}\right]\delta\mathbf{C} \tag{18}$$

where

$$\delta\mathbf{C} = -\alpha\left[\partial J/\partial C\right] + \beta \tag{19}$$

IV. Application to Tumbling Geometrical Constraints

This section introduces the application of the ASTRO theory on inspector spacecraft to inspect scientific and communication satellites that are operating in Low Earth Orbits (LEO) and Geostationary Orbits (GEO), respectively. A SPHERE-like vehicle[1] is used as the inspector spacecraft that under goes a optimally planned trajectory near the target satellite. The target satellite constraint field and motion envelope shall be defined by Sec. A and Sec. B respectively.

A. Geometrical Convex Constraint Models

To apply the ASTRO theory to a tumbling complex geometrical constraint, the target satellite must first be modeled as an assembly of individual, elementary convex constraints, such as spheres and cylinders. For on-board implementation reasons, the number of elementary constraints used to represent the actual shape of the target satellite must be limited, while maintaining sufficient geometrical details for the guidance law to be practically relevant. As an example, Fig. 1 illustrates a SSL-FS1300 like geostationary communication satellite modeled with different number of elementary constraints. The spherical constraint function is a measure of distance from which the inspector spacecraft penetrates the spherical body, and in the cylinder, the end cap constraint measures how much the inspector violates a small dome on the cap of the cylinder and the inspector penetration distance to the cylinder's radial boundary. Mathematically, the related constraint functions for a sphere f_{c_S} is defined as

$$f_{c_S}^2 = r^{-2}\left(\mathbf{x} - \mathbf{x}_c\right)^T\left(\mathbf{x} - \mathbf{x}_c\right) - 1 \tag{20}$$

where r is the radius of the sphere, $\mathbf{x}$ is the positional state of the inspector spacecraft, and $\mathbf{x}_c$ is the center of the sphere. The cylinder constraint is categorized into two types, above and below the end caps and in

between end caps. While the inspector spacecraft is above or below the end caps, the constraint f_{cC1} is defined as

$$f_{cC1}^2 = (\mathbf{x} - \mathbf{x}_c)^T \mathbf{R}^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \left(\frac{r}{l}\right)^2 \end{bmatrix} \mathbf{R} (\mathbf{x} - \mathbf{x}_c) - r^2 \tag{21}$$

where the center of the closest end cap defined as $\mathbf{x}_c$, l is the length of the cylinder, and r is the radius of the cylinder. The rotation matrix $\mathbf{R}$ is defined by Euler's theorem.[20]

$$\mathbf{R} = \mathbf{1} - \sin\theta \mathbf{a}^\times + (1 - \cos\theta)\left(\mathbf{a}\mathbf{a}^T - \mathbf{1}\right) \tag{22}$$

where $\mathbf{1}$ is the identity matrix, $\mathbf{a}$, θ, $\mathbf{l}$ and $\mathbf{k}$ are respectively defined as

$$\mathbf{a} = \frac{\mathbf{l}^\times \mathbf{k}}{\left\|\mathbf{l}^\times \mathbf{k}\right\|} \tag{23}$$

$$\theta = \arctan\left(\frac{\mathbf{l}^\times \mathbf{k}}{\mathbf{l}^T \mathbf{k}}\right) \tag{24}$$

$$\mathbf{l} = \frac{\mathbf{x}_{c2} - \mathbf{x}_{c1}}{\left\|\mathbf{x}_{c2} - \mathbf{x}_{c1}\right\|} \tag{25}$$

$$\mathbf{k} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{26}$$

where $\mathbf{x}_{c1}$ and $\mathbf{x}_{c2}$ are the positions of the end cap centers. If the inspector spacecraft position is in between the two end caps, then f_{cC2} is defined as

$$f_{cC2}^2 = \frac{\left[(\mathbf{x}_{c2} - \mathbf{x}_{c1})^\times (\mathbf{x} - \mathbf{x}_{c1})\right]^T \left[(\mathbf{x}_{c2} - \mathbf{x}_{c1})^\times (\mathbf{x} - \mathbf{x}_{c1})\right]}{(\mathbf{x}_{c2} - \mathbf{x}_{c1})^T (\mathbf{x}_{c2} - \mathbf{x}_{c1})} - r^2 \tag{27}$$

where $\mathbf{x}^\times$ is the skew-symmetric matrix of $\mathbf{x}$.[20]

B. Motion Envelope

To avoid collision with the tumbling target satellite, the inspector spacecraft plans an optimal path around a region traced out by the target satellite. As the target satellite translates and tumbles about its center of mass in six degree-of-freedom, boundary points of each elementary constraints trace out a volume in space referred to as the *motion envelope*. Six points are used to define the outer boundary points for the sphere and eight points for the cylinder as shown in Fig. 2. Given the initial states of the target satellite, all boundary points are then propagated forward in time according to Euler's equations of motion[20] for the attitude motion, and with a second-order integrator, independent on each axis, for the translational motion. The environmental torques and forces were not considered since the time interval for the inspector spacecraft to travel from the initial to the final position is brief when compared to the frequency of the environmental forces and torques. Hence, as a first order approximation, these forces and torques can be neglected and the target satellite motion can be considered to be constant. The propagation time interval is first made to be long enough to capture at least one complete period of the rotational motion. Then, the guidance law re-propagates forward in time all boundary points over a shorter time interval to obtain a smaller motion envelope. Specifically, this smaller time interval corresponds to the period of time the inspector spacecraft is expected to spend inside the original motion envelope, by assuming a straight line path at constant velocity between the two boundary conditions. An illustration of the full motion constraint field resulting from a three-element constraint target satellite rotating about two different axes is provided in Fig. 3.
The time interval of intersection between the inspector spacecraft and the motion envelope is used to determine the total number of geometrical constraints to be considered by the guidance algorithm. Ultimately, this will result in a trajectory that avoids all possible collisions while the inspector spacecraft remains as close as possible to the tumbling target satellite. Additionally, spatial margins are added to the geometrical constraints to take into account differences between the initial straight line approximation (employed in determining the integration time interval) and the final trajectory. The pseudo-code summarizing the new proximity operation optimal guidance law applicable for tumbling geometrical constraints is provided below,

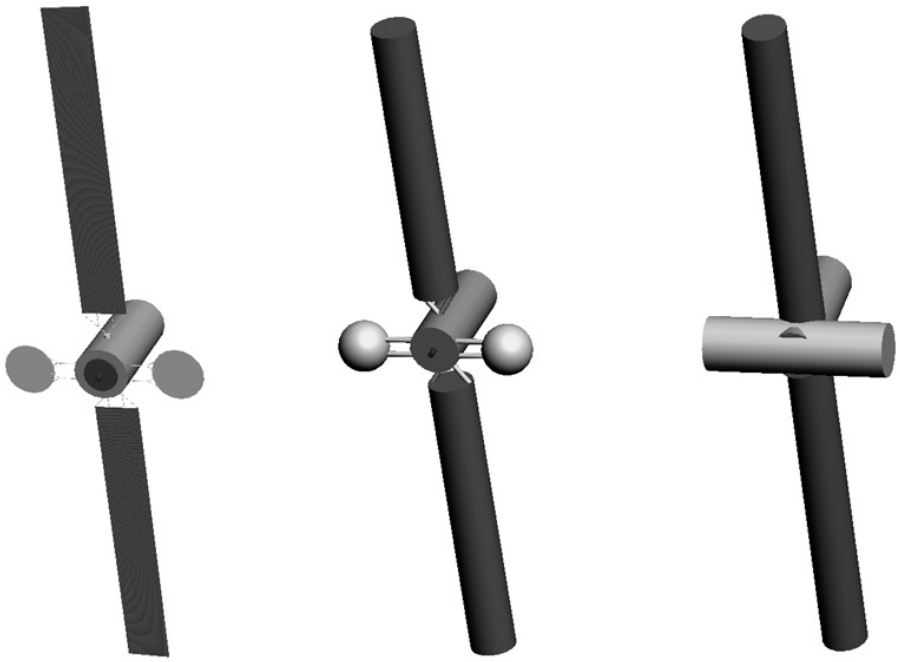

Figure 1. Target satellite modeled with 196 (left), 14 (center), and 3 (right) elementary geometrical convex constraints.

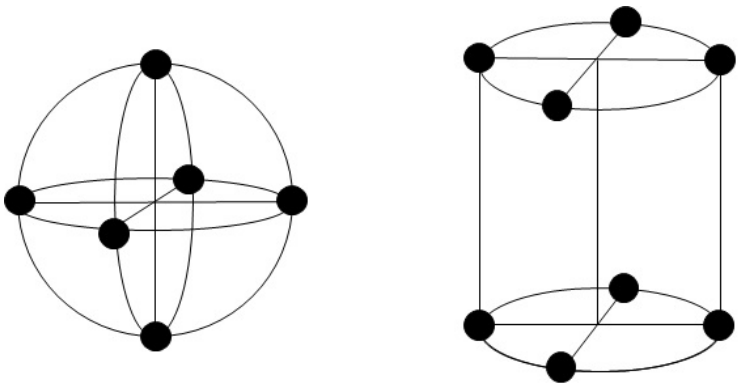

Figure 2. Boundary points of elementary geometrical convex constraints.

```
Begin
  1.0 Initialize inspector and target constraints
  2.0 Constraint Prediction
    2.1 Compute target satellite translation and rotation motion
    2.2 Compute full constraint region using target motion and boundary points
    2.3 Construct possible collision zone
    2.4 Compute entry and exit time of possible collision zone
    do While (change in entry and exit time > tolerance)
      2.5 Select new target satellite kinematics based on entry and exit time
      2.6 Repeat Steps 2.2 to 2.4
    end do
  3.0 Optimization Initialization
    3.1 Compute boundary condition matrix P
    3.2 Compute polynomial coefficient matrix C
    3.3 Compute cost function J and gradient dJ/dC
  4.0 Optimization (Outer-loop)
    do while
      4.1 Computer dC using Hessian
      4.2 Compute Gradient projection
      4.3 Compute cost function iteration, exit if dJ < tolerance
      4.4 Check dJ direction
      4.5 Inner-loop
        do while
          4.5.1 Compute trajectory
          4.5.2 Compute cost function J and gradient dJ/dC
          4.5.3 Loop through constraints for cost and gradient
          4.5.4 Exit if inner-loop dJ < tolerance
        end do
      4.6  Compute cost function J and gradient dJ/dC
      4.7  Compute Hessian (quasi-Newton method)
      4.8  Save data matrix
      4.9  Check J tolerance
      4.10 Initialize for next iteration
    end do
End.
```


V. Simulation Results

This section presents and discusses numerical simulation results obtained by implementing the aforementioned guidance law in MATLAB. Specifically, the related software is organized into input, algorithm, and output sections. During every simulation run, the input and output data is saved in .dlt and .log files, while figures and data are stored in .fig and .mat files respectively. The program also outputs an ASCII text based .dat file that is used for animation visualization in 3D Studio Max. The ASTRO software engine which was initially developed for ISS SPHERES testbed operations, was revised for this work so it could be combined with the front-end satellite dynamics motion propagation. Three vehicle types are simulated, these are: a SPHERES[16] class inspector spacecraft, the SSL-FS1300 class geostationary target satellite, and a Envisat class scientific target satellite. The inspector spacecraft is paired with either SSL-FS1300 or Envisat in each simulation case. The envelope satellite geometries used in simulations are provided in Table 1.

Table 1. Inspector spacecraft and target satellite geometric envelopes.

Description	SPHERES	SSL-FS1300	Envisat
Vehicle height (m)	0.125	31.27	10.00
Vehicle width (m)	0.125	8.64	28.33
Vehicle depth (m)	0.125	6.11	7.88

Regardless of its finite dimensions, in the guidance law, the inspector spacecraft is modeled as a point mass,

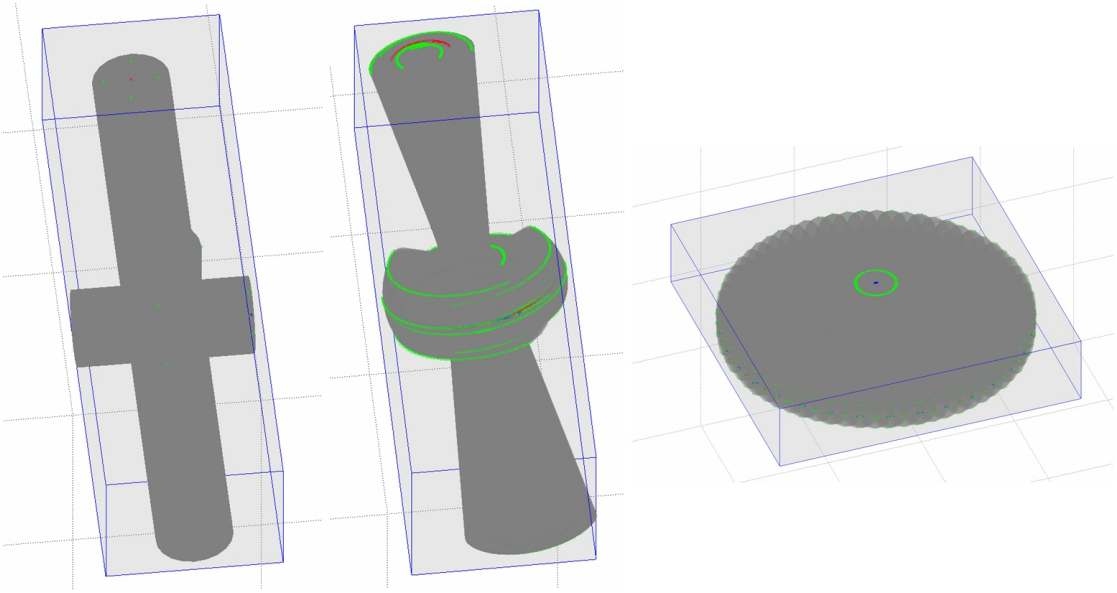

Figure 3. Motion envelope traced by boundary points of elementary geometrical constraints.

which requires the target satellite geometry to incorporate the dimensions of the inspector spacecraft. As shown in Fig. 4 and Fig. 5, the inspector spacecraft geometry is therefore accounted for by adding a safety buffer of the inspector radius to the target satellite constraint field, a 100% radius margin of inspector diameter is also added to the buffer for additional safety.

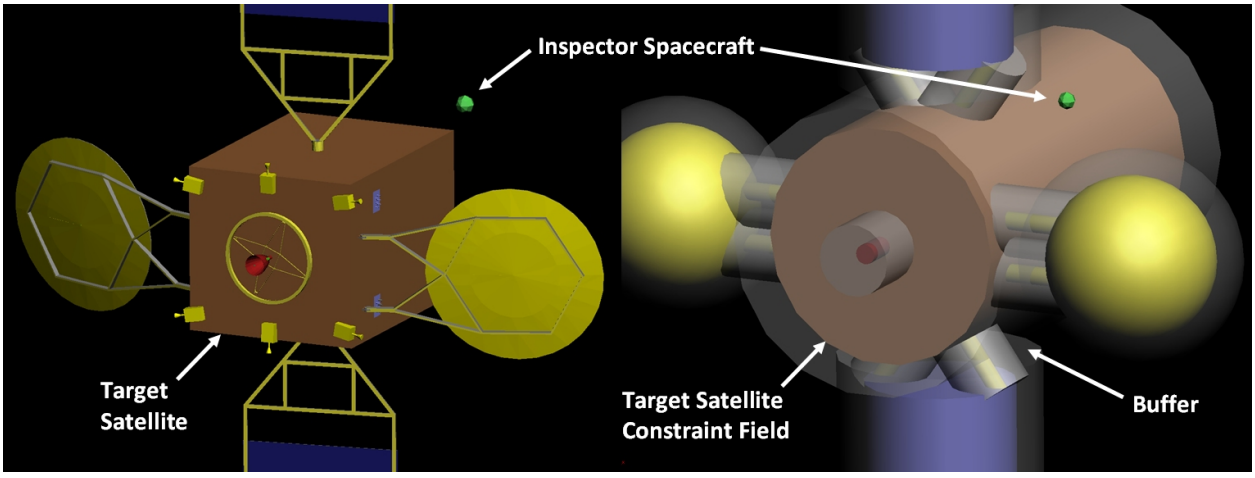

Figure 4. Geometric buffer surrounding SSL-FS1300 satellite.

For all simulation cases defined below, the boundary condition functions of the inspector spacecraft are defined as

$$f_{BC_1}\left(\begin{bmatrix} 0 & 0 & -10 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}\right) = 0$$
$$f_{BC_2}\left(\begin{bmatrix} 0 & 0 & 10 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}\right) = 0$$

Thus, the inspector spacecraft is initially positioned on the negative R-bar (zenith-direction), where it can

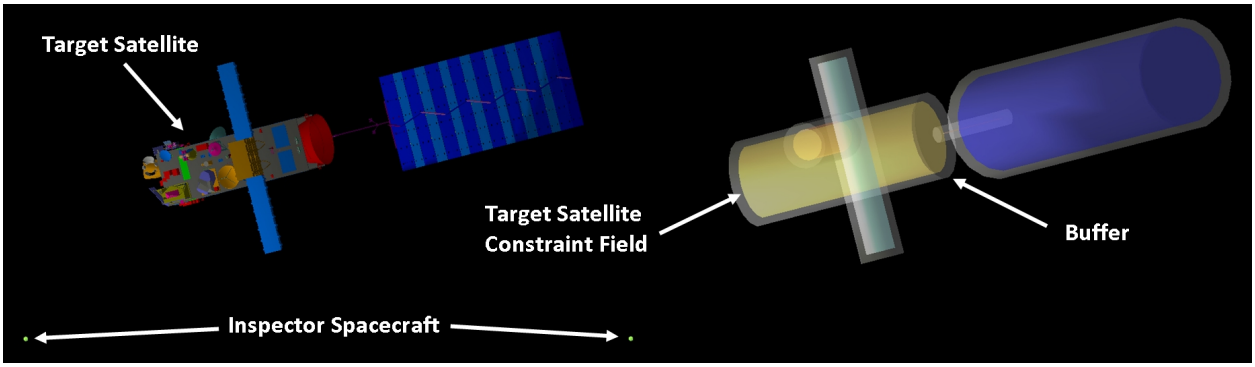

Figure 5. Geometric buffer surrounding Envisat satellite.

would observe the target satellite motion and plan the safest optimal path using the proposed guidance law. The inspector spacecraft translates past the target satellite to positive R-bar (nadir-direction).

Multimedia extensions are attached to this paper.[b] In these videos, the bottom graphs emulate the left and right camera view of a pair of grayscale stereo cameras installed on the inspector spacecraft. These two graphs were generated by assuming that both cameras are always pointing at the target satellite. The top right graph initially shows a camera view that is aligned with the velocity direction. When the target satellite gets outside the field of view, this velocity direction-poiting camera view is replaced by a three-dimensional graph that illustrates different paths obtained at different iterations of the algorithm (in yellow) as well as the final, optimal path (in red). Finally, the top left graph illustrates the inspector spacecraft following the desired path, as calculated by the guidance law, in close proximity to the translating and tumbling satellite.

A. Case 1: Stationary Target Satellite

Simulation results illustrating the optimal solution around a stationary satellite are provided in Fig. 6. Where the solid dot represents the initial position of the inspector spacecraft and the hollow dot represents the final position. This simulation demonstrates the core ASTRO algorithm's capability in determining the optimal path after 48 outer-loop iterations. It should be noted for this simulation, a model with 3-elementary constraints was used in the determination of the optimal trajectory.

B. Case 2: Tumbling and Spinning Target Satellite

The target satellite motion is based on the tumbling rotational rate of a GOES 8 geostationary satellite, as observed by Cognion et al.,[21] and the rotational rate of Envisat as observed by Kurcharski et al.[22] Specifically, a drift rate of -0.01 m/s in the x-axis and a tumbling rate of 0.28, 2.8, 0.28 deg/s about the roll, pitch, and yaw axis is used for a slow spin simulation, and 21.8 deg/s about the yaw axis is used for a fast spin simulation. For the slow tumbling target satellite, simulation results for the number of iterations before reaching a feasible solution around a tumbling geostationary satellite are provided in Fig. 7. Based on the solution feasibility of achieving the constraint objective, a sub-optimal solution is achieved after 32 outer-loop iterations; the optimal solution however was not reached until after 133 outer-loop iterations. The lower fidelity three body model was used for this simulation in order to minimize the the high number of total bodies to be considered for motion over the entire time period. The total time for reaching the optimal 133 iteration is 3.0 times the stationary simulation result for SSL-FS1300, which is herein defined as t_{fs}. If the 32-iteration sub-optimal feasible solution is forced by the user manually, the total computation time is reduced to $0.8t_{fs}$. For the fast spinning target satellite, simulation results are provided in Fig. 8. The optimal solution is achieved after 52 iterations, the total simulation time is $1.4t_{fs}$.

[b]Data available online at https://www.youtube.com/watch?v=hlSqW3wOjiU&list=PLeAC9P53bklfk1CbkL6zI7zKip6_g-Lg7 [retrieved 25 January 2016].

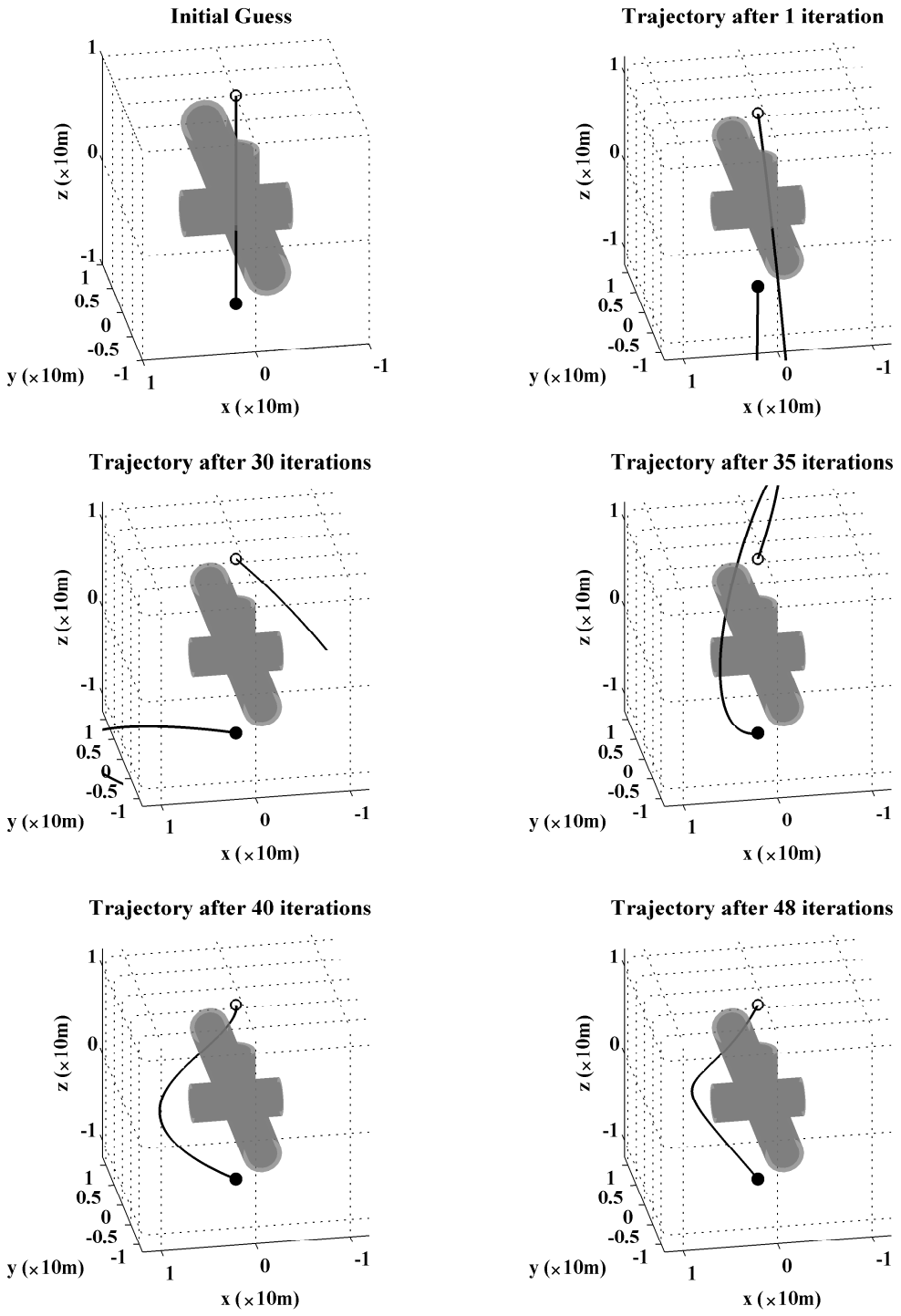

Figure 6. Optimal path around a stationary geostationary satellite.

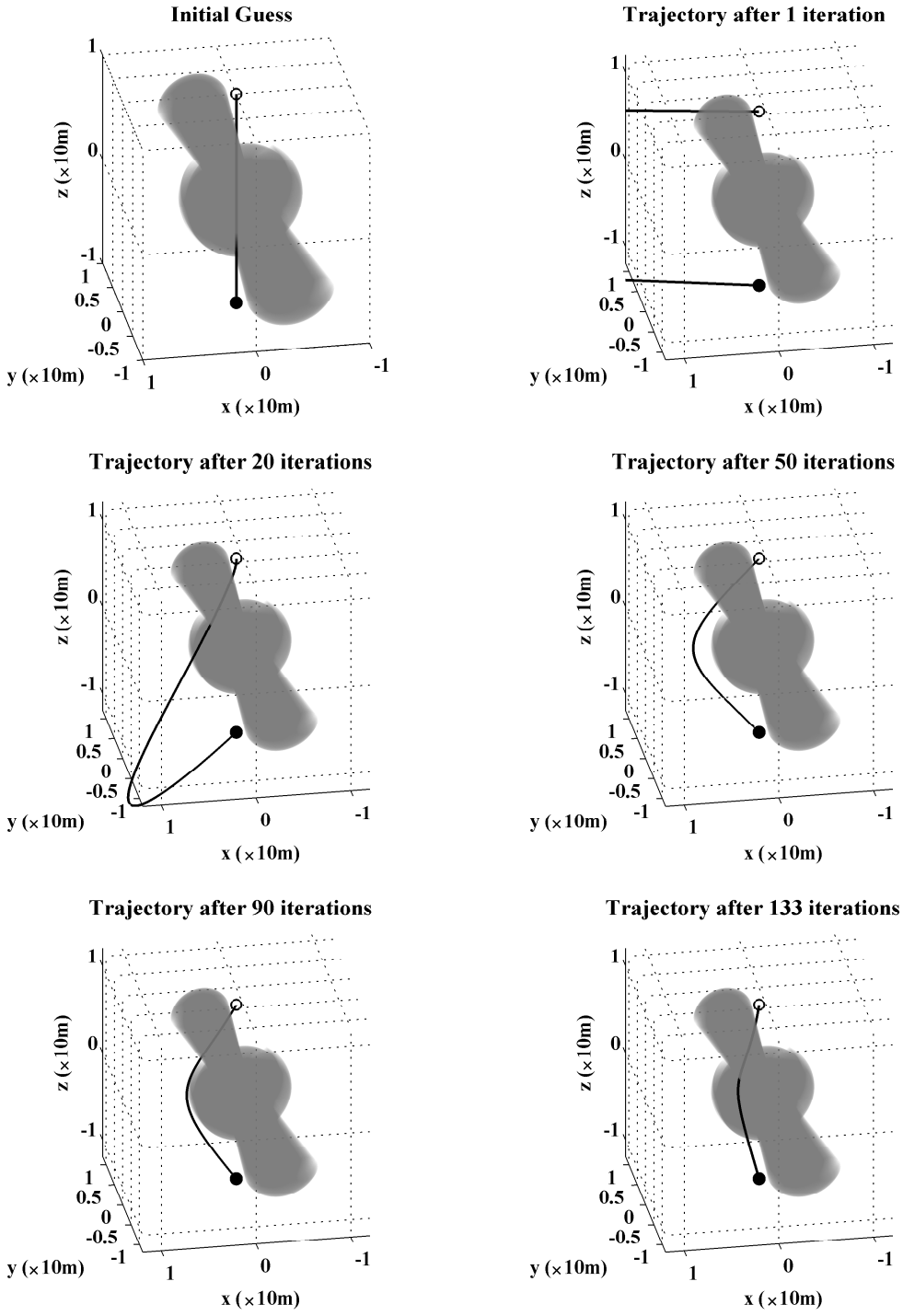

Figure 7. Optimal path around a slow tumbling geostationary satellite.

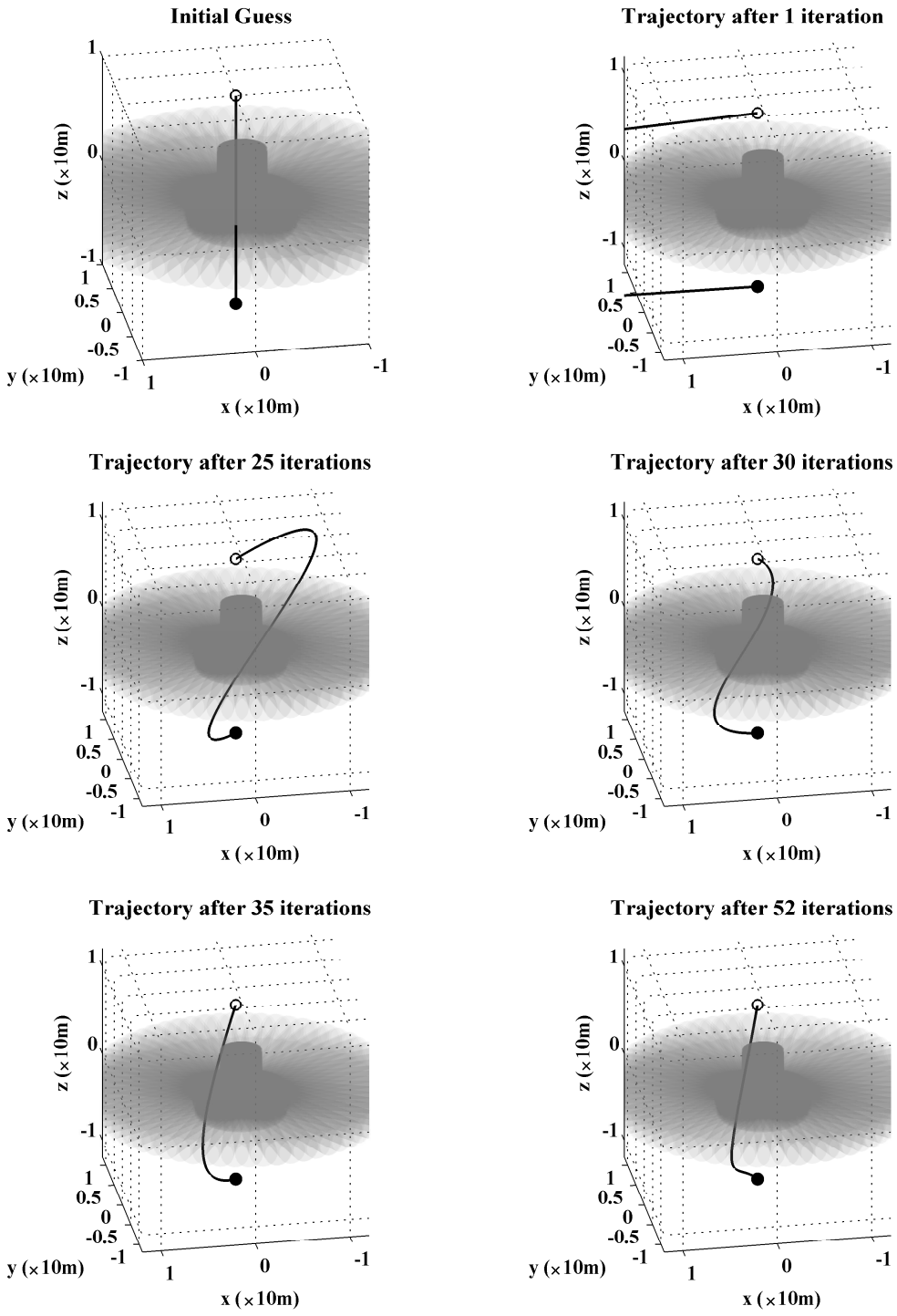

Figure 8. Optimal path around a fast-spinning geostationary satellite.

C. Number of constraints and polynomial order

The Legendre Polynomial order and number of constraint body fields can be increased to achieve desired fidelity in the simulation solution at the cost of higher computational time. The baseline simulation cases defined above use 6th-order polynomials and three constraint bodies. To analyze the effects of the polynomial order and number of constraint bodies on the ASTRO algorithm, these parameters are varied up to 12th-order polynomials and 14 constraint bodies. The trajectory solution and solution feasibility simulation results of higher polynomial order and constraint bodies are shown in Fig. 9 and Fig. 10 respectively. In this particular example, results suggest that lower-order polynomials combined with and a simpler geometrical constraint achieve a more direct path than the higher-order polynominals. However, in general, the more complex the geometrical constraints are, the more complex the desired path needs to be, in turn, requiring higher-order coeficients. On the other hand, if a simple path suffices, then higher-order polynomials will only result in more iterations to find a solution.

D. Constraint Corridor

The initial iteration typically provides a path that is highly eccentric, compared to the final optimized solution. Indeed, this first iteration corresponds to the lowest-order solution that satisfies the boundary conditions, without violating the geometrical constraints. However, such a long and eccentric path is not practical in terms of time required to follow it, nor in terms of spacecraft fuel usage. To speed up the convergence and to bound feasible solutions to practical ones, an outer boundary corridor can be used to restrict highly eccentric solutions. Such a boundary corridor uses the inside of the cylindrical constraint to produce an opposite effect of a typical constraint body. An illustration of the trajectory solution and solution feasibility for with and without the constraint corridor is shown in Fig. 11 and Fig. 12, respectively. The trajectory solution for the bounded constraint corridor is much closer to the target satellite body and the second iteration cost for the unbounded solution is 724 times the corridor bounded solution. The corridor solution converged at 60 iterations and is $1.5t_{fs}$. In addition, a constraint corridor-bounded solution produces a practically feasible initial solution to be tracked when the on-board computational resources are limited. It should be noted however, the corridor constraint approach can sometimes fail to generate a solution. Further investigations in optimizing of the constraint corridor approach are required to increase its robustness.

E. Outer Geometry Replacement

Further improvements can be made to the computational time by simplifying the total number of bodies in the entire motion region since many of the bodies occupy redundant volumes. This can be done by analyzing the constraint field over the entire motion time and trace out a single avoidance region. Replacing the grouped constraint field with a single, stationary constraint body results in a simple target satellite region for the guidance law to solve. An example of the geometry replacement method is shown by Fig. 13 and Fig. 14 for trajectory solution and solution feasibility respectively. As shown in Fig. 14, using a single constraint allows the algorithm to quickly converge after 46 iterations, with $0.005t_{fs}$, compared to $3.0t_{fs}$ for the baseline simulation. Although the outer geometry replacement method forces the inspector spacecraft to travel further away from the target satellite, the improvement in performance from the geometry replacement is significant. This method can be used when computational resources are scarce and real-time computational ability is required.

F. Target Satellite: Envisat

The Envisat is simulated based on satellite tumbling data provided by Kurcharski et al.[22] Similar to the SSL-FS1300, a stationary, slow tumble, and fast spin simulation cases were performed using the same rates as defined previously. Figure 15 and Fig. 16 shows the stationary and fast spin trajectory solution and solution feasibility respectively. The stationary Envisat optimal solution requires 56 outer loop iterations while the fast spin solution requires 40 outer loop iterations. The simulation time for the fast spin solution is 414 times the stationary solution result for Envisat, herein defined as t_{es}. The baseline initial state resulted in the boundary condition to be inside the motion constraint field of Envisat. A comparison of the baseline initial state with a initial state twice the distance of the baseline was performed to determine the behaviour when boundary conditions are inside the motion envelope. Figure 17 and Fig. 18 shows the slow tumbling Envisat trajectory solution and solution feasibility for inside and outside the Envisat constraint field, respectively.

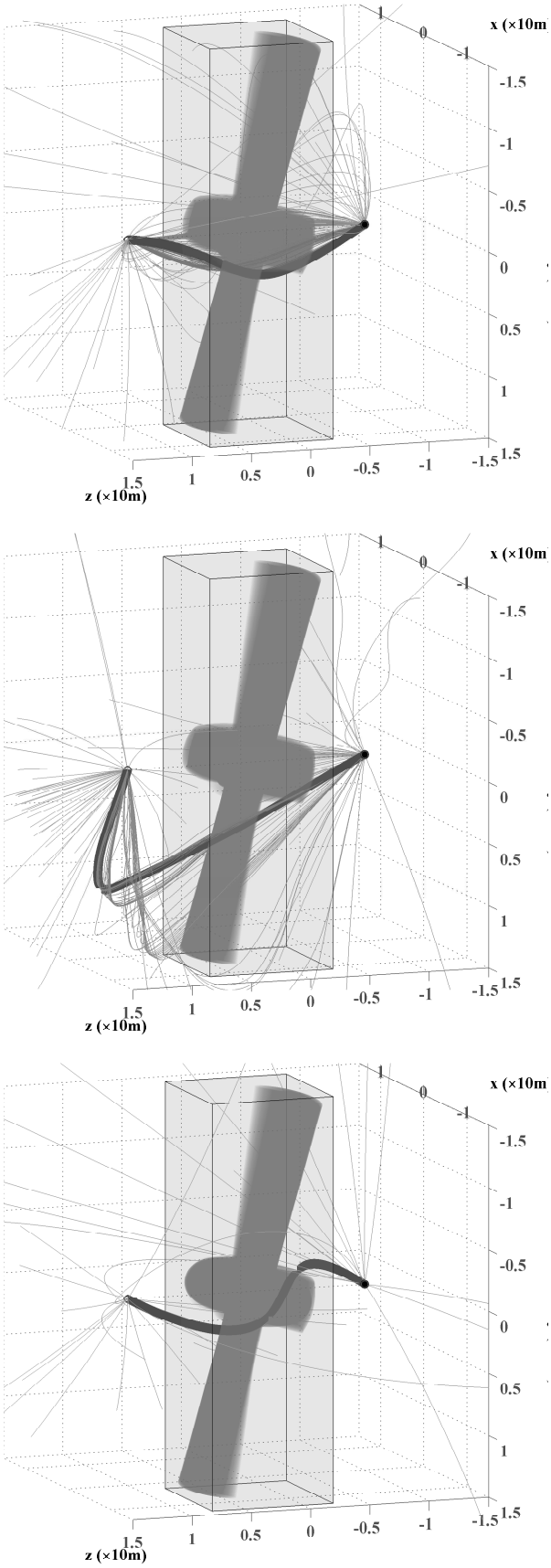

Figure 9. 3 constraint 6 deg (top), 3 constraint 12 deg (center), and 14 constraint 6 deg (bottom).

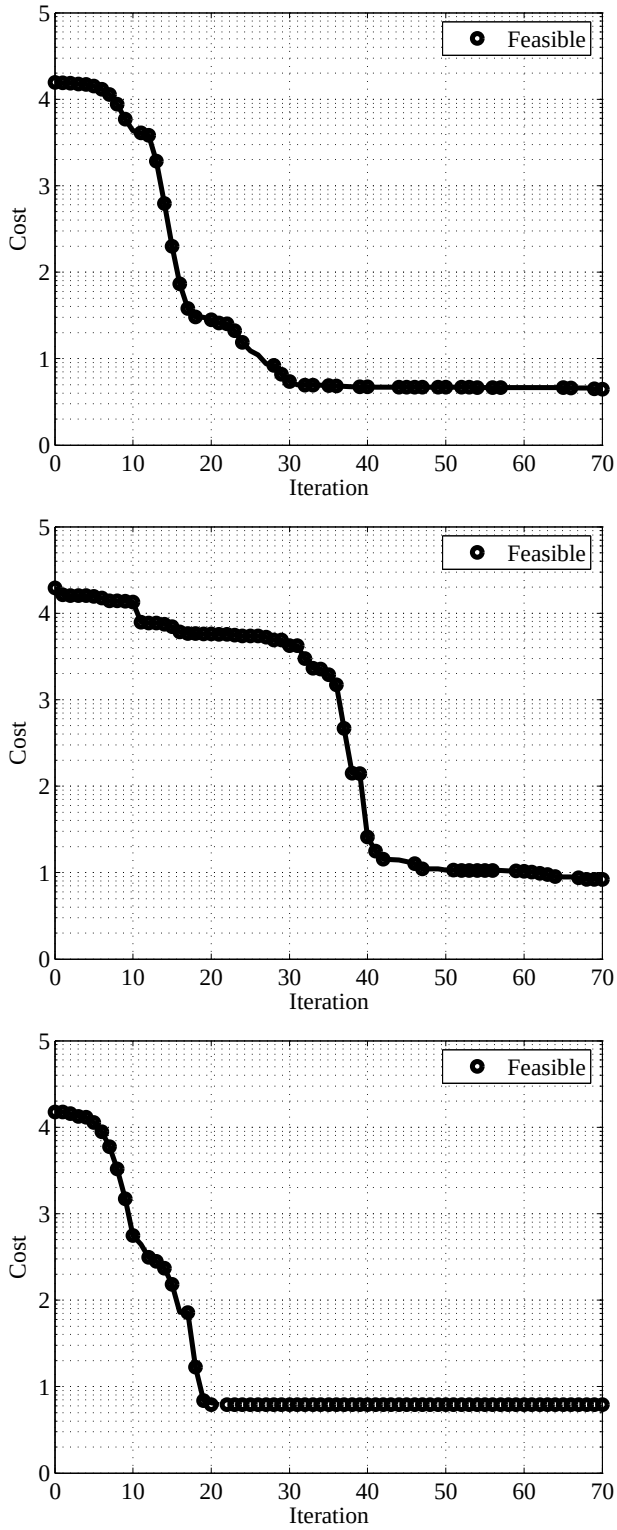

Figure 10. Feasibility graphs. 3 constraint 6 deg(top), 3 constraint 12 deg(center), and 14 constraint 6 deg(bottom).

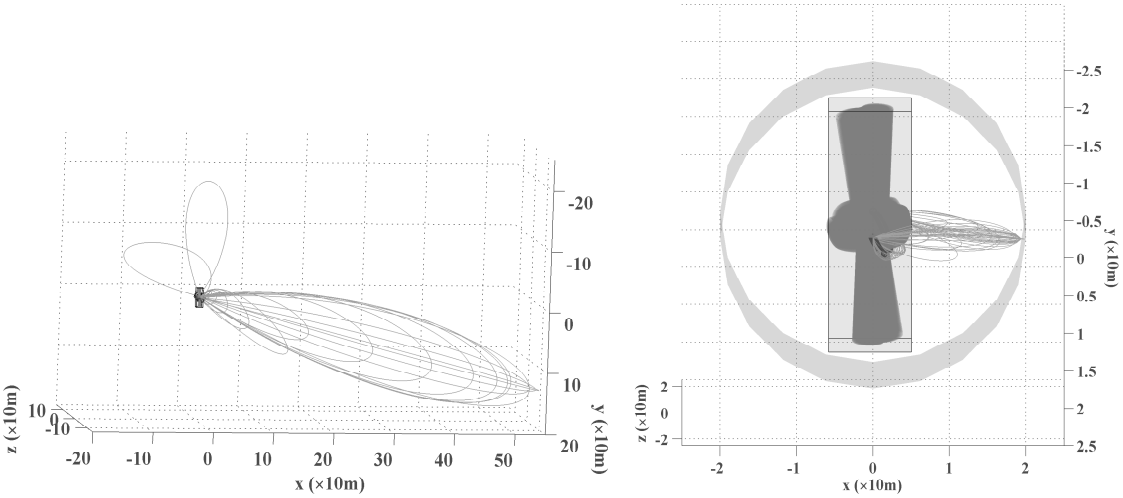

Figure 11. Without (left) and with (right) constraint corridor.

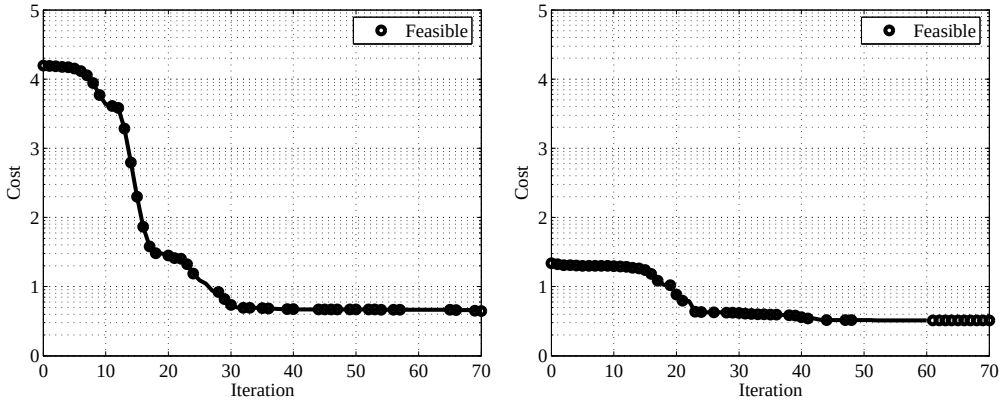

Figure 12. Feasibility graphs. Without (left) and with (right) constraint corridor.

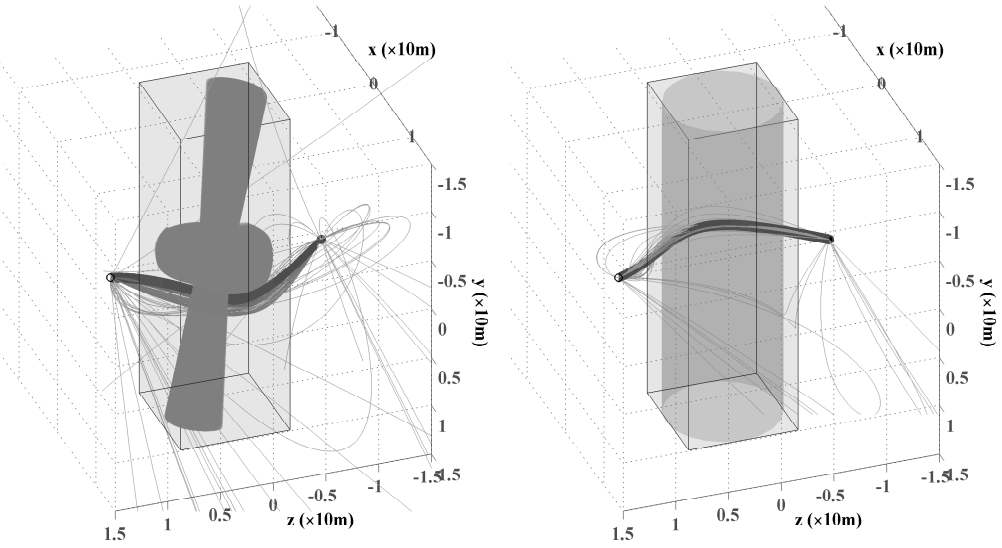

Figure 13. 3-body (left) and 1-body (right) constraint field.

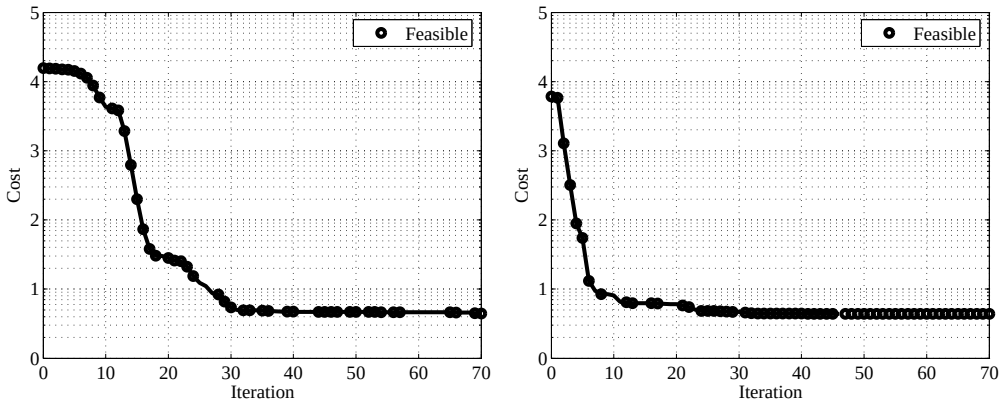

Figure 14. Feasibility graphs. 3-body (left) and 1-body (right) constraint field.

In the case where boundary conditions are inside the constraint field, an optimal solution is achieved after 219 iterations at the cost of $1717t_{es}$, this is compared to 59 iteration and $255t_{es}$. Although the feasibility graphs shows that there is no feasible solution inside the constraint field, it should be noted that the motion envelope is a conservative assumption of all possible constraints occurring at any time. In reality, certain combinations of target satellite constraints and inspector spacecraft position can still produce a non-colliding solution. Such combinations can be subject to future studies of the ASTRO algorithm. Without a reliable way of determining these combinations, boundary condition must therefore be limited to staying outside the full constraint field for maximum safety.

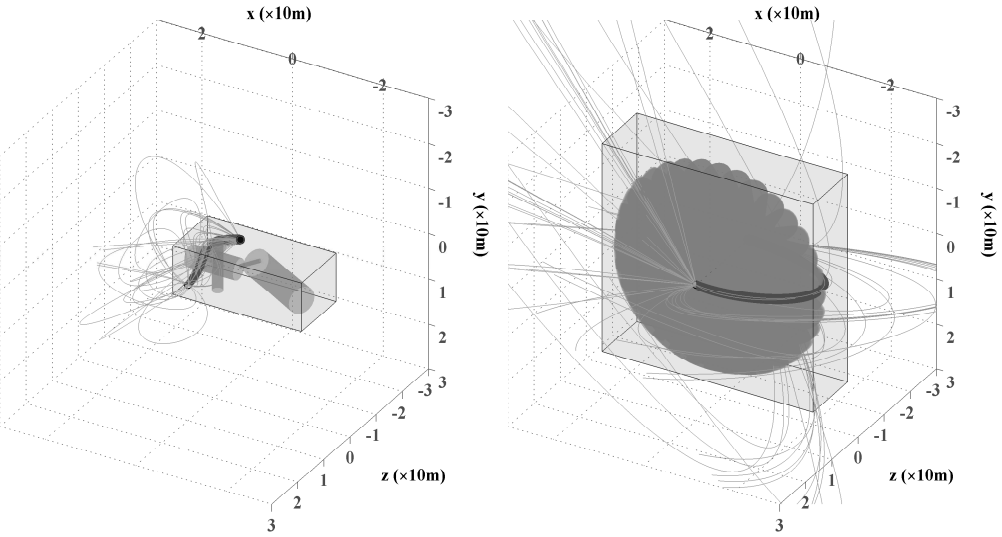

Figure 15. Envisat stationary (left) and fast spin (right) constraint field.

VI. Conclusions

This paper examined the Admissible Subspace Trajectory Optimizer guidance law and provided modifications to the algorithm to accommodate rigid body motion in translation and rotation. Specifically, the target object was first modeled as an assembly of individual, elementary convex constraints, such as spheres and cylinders. Then, once the object is properly modeled, the guidance law propagates forward in time all boundary points of this object to obtain a high-risk collision region in three-dimensional space referred to as the *motion envelope*. Finally, the Legendre polynomial-based proximity guidance law iteratively computes a sub-optimal solution, then with available computation time, an optimal, admissible trajectory to be followed

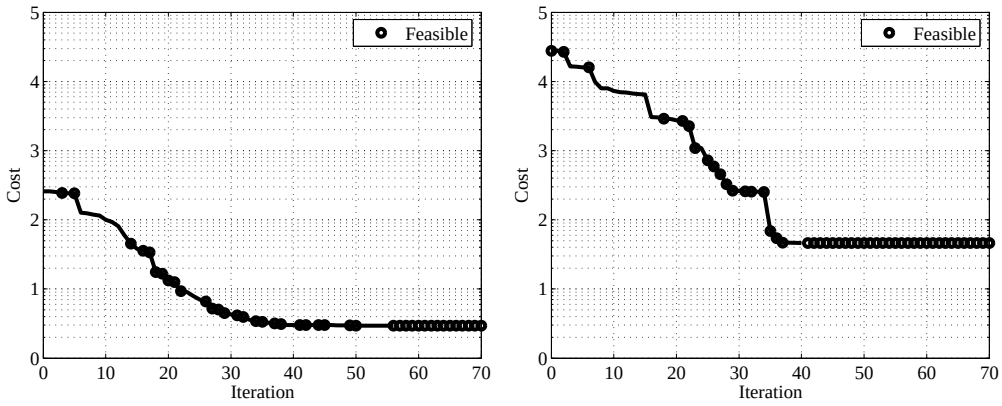

Figure 16. Feasibility graphs. Envisat stationary (left) and fast spin (right) constraint field.

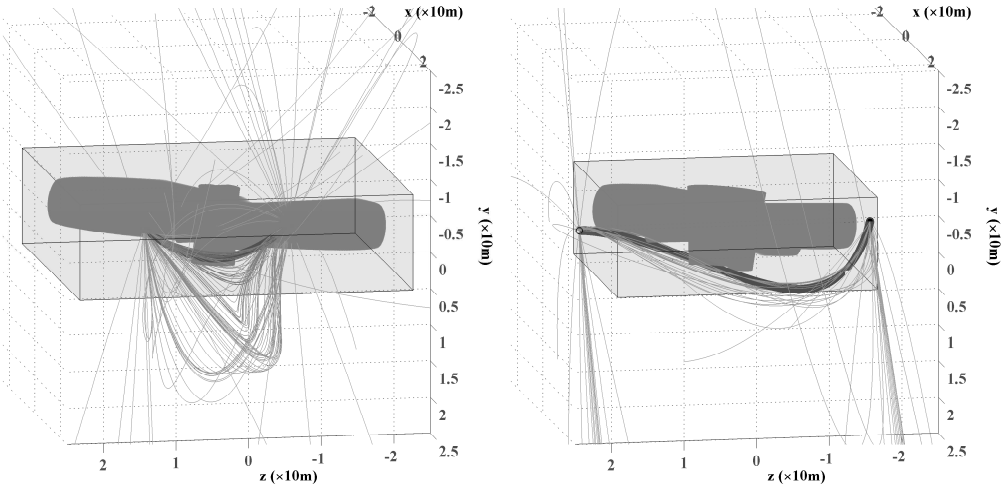

Figure 17. Slow tumbling Envisat approach inside (left) and outside (right) constraint field.

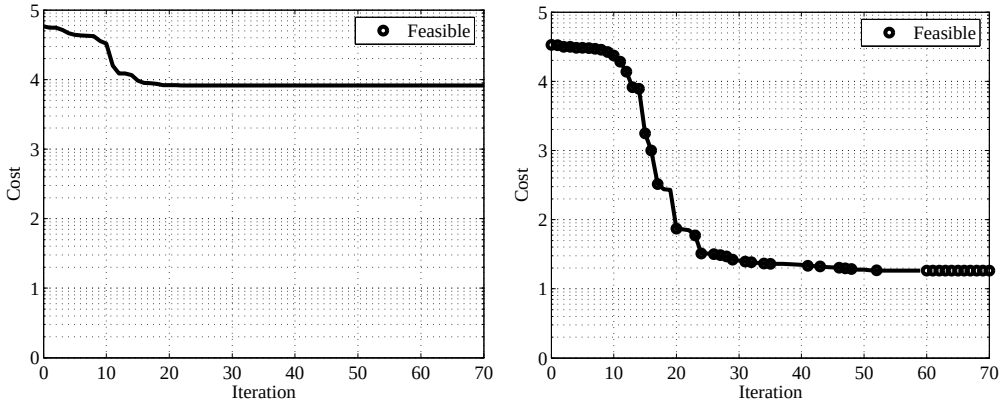

Figure 18. Feasibility graphs for slow tumbling Envisat approach inside (left) and outside (right) constraint field.

by the inspector, which avoids the motion envelope. The performance of the guidance law was evaluated in numerical simulations for inspection maneuvers around a stationnary, a slow tumbling and a fast spinning Envisat spacecraft and SSL-FS1300 geostationary communication satellite. Additionally, different orders of the path defined by the Legendre polynomials and number of constraints were compared. Future work will focus on performing extensive experimental validations in a three-DOF environment using free-floating air bearing platforms at Carleton University, as well as comparing the performance of the proposed guidance law against existing ones, such as the Probabilistic Sulu (p-Sulu) algorithm and simple cubic, quartic and pentic spline-based trajectories.

Acknowledgments

This research was partially funded by the Natural Sciences and Research Council of Canada's Engage Grant award EGP #469958-14 and the Alexander Graham Bell Canada Graduate Scholarship CGS D3-453738-2014. The authors also acknowledge the technical contributions of Alvar Saenz-Otero, and Jakob Katz from the Massachusetts Institute of Technology for the implementation of the original ASTRO guidance law and on-orbit testing. Special thanks to Stephane Ruel of Neptec Design Group Ltd. for providing the Envisat CAD model used in this work.

References

[1]Fourie, D., Tweddle, B. E., Ulrich, S., and Saenz-Otero, A., "Flight Results of Vision-Based Navigation for Autonomous Spacecraft Inspection of Unknown Objects," *Journal of Spacecraft and Rockets*, Vol. 51, No. 6, 2014, pp. 2016–2026.

[2]Taur, D. R., Coverstone-Carroll, V., and Prussing, J. E., "Optimal Impulsive Time-Fixed Orbital Rendezvous and Interception with Path Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 1, 1995, pp. 54–60.

[3]Munoz, J. and Fitz-Coy, N., "Rapid Path-Planning Options for Autonomous Proximity Operations of Spacecraft," *AIAA Guidance, Navigation, and Control Conference*, Toronto, ON, 2010, AIAA paper 2010-7667.

[4]Ulybyshev, Y., "Trajectory Optimization for Spacecraft Proximity Operations with Constraints," *AIAA Guidance, Navigation, and Control Conference*, Portland, OR, 2011, AIAA paper 2011-6629.

[5]Kim, Y., Mesbahi, M., and Hadaegh, F. Y., "Dual-Spacecraft Formation Flying in Deep Space: Optimal Collision-Free Reconfigurations," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, 2003, pp. 375–379.

[6]Hadaegh, H. Y. and Singh, G., "Collision Avoidance Guidance for Formation-Flying Applications," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Montreal, QC, 2001, AIAA paper 2001-4088.

[7]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. 26, No. 4, 2002, pp. 775–764.

[8]Breger, L. and How, J. P., "Safe Trajectories for Autonomous Rendezvous of Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1478–1489.

[9]Luo, Y., Lei, Y., and Tang, G., "Optimal Multi-Objective Linearized Impulsive Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 2, 2007, pp. 383–389.

[10]Luo, Y., Lei, Y., and Tang, G., "Optimal Multi-Objective Nonlinear Impulsive Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 4, 2007, pp. 994–1002.

[11]Ranieri, C., "Path-Constrained Trajectory Optimization for Proximity Operations," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, HI, 2008, AIAA paper 2008-6275.

[12]Alizadeh, F. and Goldfarb, D., "Second-Order Cone Programming," *Mathematical Programming*, Vol. 95, No. 1, 2003, pp. 3–51.

[13]Boyd, S., *Convex Optimization*, Cambridge University Press, Cambridge, England, 2004.

[14]Lu, P. and Liu, X., "Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 2013, pp. 375–389.

[15]Chamitoff, G. E., Saenz-Otero, A., Katz, J. G., and Ulrich, S., "Admissible Subspace Trajectory Optimizer (ASTRO) for Autonomous Robot Operations on the Space Station," *AIAA Guidance, Navigation, and Control Conference*, National Harbor, MD, 2014, AIAA paper 2014-1290.

[16]Saenz-Otero, A. and Miller, D., "SPHERES: A Platform for Formation-Flight Research," *SPIE International Symposium on Optics and Photonics*, San Diego, CA, 2005, SPIE Code Number 5899-25.

[17]Morrell, B. J., Gibbens, P. W., and Chamitoff, G. E., "Application of a Trajectory Optimisation Algorithm for Dynamic Obstacle Avoidance and Multiple," *Fourth Australasian Unmanned Systems Conference*, Melbourne, Australia, 2014.

[18]Morrell, B. J., Chamitoff, G. E., Gibbens, P. W., and Saenz-Otero, A., "Autonomous Operation of Multiple Free-flying Robots in the International Space Station," *25th AAS/AIAA Space Flight Mechanics Meeting*, Williamsbug, VA, 2015, AAS paper 15-301.

[19]Tweddle, B. E., Ulrich, S., Setterfield, T., Saenz-Otero, A., and Miller, D., "The SPHERES-VERTIGO Goggles: An Overview of Vision-Based Navigation Research Results from the International Space Station," *12th International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, Montreal, CA, 2014.

[20]Hughes, P. C., *Spacecraft Attitude Dynamics*, John Wiley & Sons, New York, NY, 1986.

[21]Cognion, R., Albuja, A. A., and Scheeres, D. J., "Tumbling Rates of Inactive Geo Satellites," *International Astronautical Conference*, Toronto, ON, 2014, IAC paper 14.C1.2.12x25119.

[22]Kucharski, D., Kirchner, G., Koidl, F., Fan, C., Carman, R., Moore, C., Dmytrotsa, A., Ploner, M., Bianco, G., Medvedski, M., Makeyev, A., Appleby, G., Suzuki, M., Torre, J., Zhang, Z., Grunwaldt, L., and Qu, F., "Attitude and Spin Period of Space Debris Envisat Measured by Satellite Laser Ranging," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 52, No. 12, 2014, pp. 7651–7657.