OPTIMAL TRAJECTORY GUIDANCE FOR SPACECRAFT ROBOTIC SERVICING MISSIONS

Jian-Feng Shi[1], Steve Ulrich[1], and Andrew Allen[2]

[1]*Carleton University, Ottawa, Ontario, K1S 5B6, Canada*
[2]*MacDonald, Dettwiler and Associates Ltd., Brampton, Ontario, L6S 4J3, Canada*

ABSTRACT

This paper presents the development and application of a real-time optimization guidance algorithm capable of generating trajectories for close proximity operations in a three-dimensional environment which includes static and/or dynamic obstacles. Such obstacles are representative of non-cooperative, and spinning target spacecraft. The on-orbit servicing problem defined for this study requires a small free-flyer robotic spacecraft to manoeuvre in close proximity to a large non-cooperative satellite. Simulation results are provided for two mission scenarios: (1) approach and fly around of a non-cooperative, but stationary target spacecraft, and (2) approach and fly around of a non-cooperative and spinning target spacecraft. This study builds upon the previously-developed Admissible Subspace Trajectory Optimizer theory to consider tumbling spacecraft defined with basic geometric shapes. This comprehensive study provides the theory as well as simulation results.

Key words: optimal path-planning; trajectory guidance law; Legendre polynomials.

1. INTRODUCTION

Spacecraft close proximity inspection and proximity operations are important elements of on-orbit servicing (OOS). Future robotic systems with the capability to safely navigate around a moving target object using real-time autonomous decision-making algorithms without human guidance and intervention will enable a new class of space servicing missions. Due to advancements in electronic miniaturization, future OOS missions may utilize multiple nanosatellite-sized drones deployed on orbit by a single larger, host servicer spacecraft [1]. To this end, the US Defense Advanced Research Project Agency (DARPA) is currently investigating a Payload Orbital Delivery (POD) system[1], i.e., a intelligent spacecraft drone, which is being designed and built by MacDonald Dettwiler and Associates (MDA) Robotics and Automation (see Figure 1). Such autonomous drones 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 any damage in the event of accidental collision. In addition, in cases of large-scale space structure maintenance, the use of spacecraft drones could eliminate the need for various on-board cameras while minimizing fuel usage for the host servicing spacecraft.

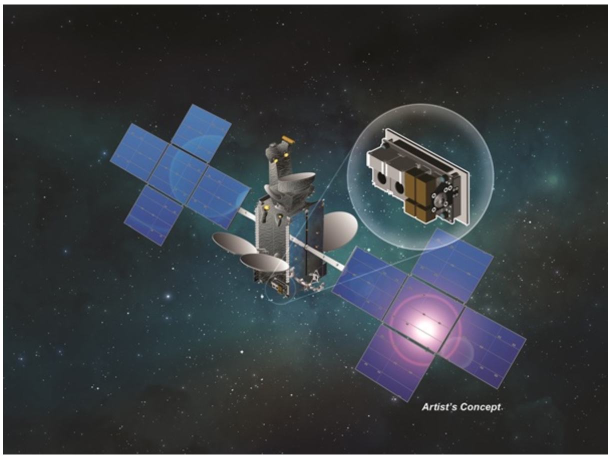

Figure 1. DARPA's Payload Orbital Delivery (POD) system.

In such drone inspection missions, an on-board guidance algorithm would be highly valuable to enable the drones to autonomously maneuver in close proximity to tumbling satellites. To adress this specific problem is the real-time trajectory 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. ASTRO transforms the problem into a parameter space that is well behaved; it can navigate complex surroundings with multiple obstacles and constraints. This spceific guidance strategy was sucessfully tested on the International Space Station (ISS) Synchronized Position Hold Engage Reorient Experimental Satel-

[1] Phoenix Release POD Interface Requirements as First Step Toward Vision of FEDEX TO GEO, http://www.darpa.mil/newsevents/releases/2014/11/10.aspx

lite (SPHERES) testbed. 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 [2].

Previous work that addressed the real-time nonlinear constrained optimization problem include a thrust limited manoeuvre with path constraint approach by Taur *et al.* [3], an adaptive artificial potential functions proposed by Munoz and Fitz-Coy [4], an optimization for spacecraft proximity manoeuvre with interior point constraints by Ulybyshev [5], a minimum relative distance constraint via indirect optimization by Hadaegh *et al.* [6], a spline trajectory by Hadaegh and Singh [7], the so-called mixed-integer linear programming (MILP) by Richard *et al.* [8], Breger and How [9], a genetic algorithm by Luo *et al.*[10, 11], a sparse optimal control software (SOCS) by Boeing [12], and second-order cone programming (SOCP) by Alizadeh *et al.* [13], Boyd *et al.* [14], and Lu and Liu [15].

During the development of the ASTRO algorithm, the multi-dimensional optimization problem for spacecraft proximity operations was studied. A signature feature of the ASTRO algorithm is the ability to provide path interior constraints rather than simply terminal constraints.The initial 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 1 second intervals. The ASTRO computation could however take up to 10 seconds to complete when computing on a command laptop [2]. This previous development only considered a singular point geometry spherical object travelling at a constant velocity within cylindrical corridors. In addition, the geometrical constraints did not provide geometric pose restrictions. In this context, this paper bridges the gap between the use of singular spherical constraints and a more complex and realisitc, spinning geometrical constraint which is representative of a tunbling uncooperative satellite. Additionally, improvements to the algorithm to maximize its efficiency in computational speed, so it is better-suited for real-time operations, are proposed.

This paper is organized as follows: Section 2 defines the autonomous trajectory guidance problem. Section 3 summarizes the theoretical development of the original ASTRO algorithm and described the modifications proposed herein to accomodate rotating geometrical constraints. Section 4 describes the MATLAB simulation environment and provides an overview of the software program architecture. Section 5 provides simulation results demonstrating the performance of the modified ASTRO algorithm for a servicer drone spacecraft navigating around both a stationary and an uncooperative rotating geostationnary satellite. Finally, Section 7 summarizes this work.

2. PATH-PLANNING OBJECTIVE

The objective of the guidance algorithm is to compute a feasible trajectory for a spacecraft drone around a large and complex geometric object that is both translating and rotating in space. Geometric constraints, such as those imposed by the object's solar arrays sweeping out a region of space, should be avoided by the spacecraft drone. On the other hand, the path computed by the guidance law for the spacecraft drone should be minimized for fuel saving purposes. Since the modification to the ASTRO algorithm is focused on the path planning aspect of trajectory guidance, it is assumed the spacecraft drone has already performed a visual inspection of the target satellite for a sufficiently-long time and, as such, has the ability to propagate the dynamics of the target object forward. The propagation time is bounded by the initial guess of overall translational maneuver time of the servicer drone, i.e. from 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 such as mass properties and thrusters limitations. The second step attempts to achieve an optimal trajectory that also minimizes the path length.

3. MODIFIED GUIDANCE ALGORITHM

The modification to the original ASTRO algorithm provides initialization in determining the appropriate geometric constraint for ASTRO solver planning. During this initialization, the algorithm evaluates the geometric body of the target satellite object and its motion. Specifically, the modified ASTRO algorithm can be broken down into three steps:

1. the prediction of the motioning geometric body target;
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.

Each of these three steps are described in details in the following section.

3.1. Target Body Geometry Prediction

There are two ways to perturb bodies in the guidance algorithm. The first is to do it on a individual sub-body

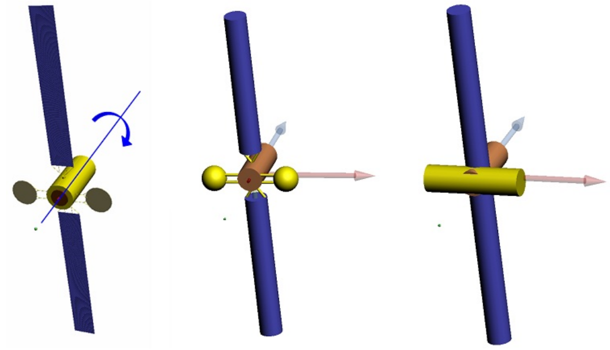

Figure 2. Target satellite model at various resolutions.

level, this allows a new cost to be computed at each iteration due to body motion. The second is to do it at a global level, this is the method employed in this paper. Although at the individual level one could program complex cost motion, this however can only be done with simple geometries. Once the simple geometries are build up to larger assemblies, the motion becomes too complex to be programmed at individual sub-body level. A constant rigid body dynamic motion is used to demonstrate this concept; however, it can be extended to flexible nonlinear translation and rotation without restrictions.

Constraint functions of general obstacles can be constructed by constrained volume (walls), ellipsoid or more complex convex shapes such as cylinders, planes of any orientation, maximum speeds or accelerations, maximum path curvature, and moving obstacles. For total geometry constraints, spheres and cylinders will be used as baseline elemental geometries. For practical computational resource considerations, the target satellite must be restricted to a limited number of elementary bodies (shapes) to represent the actual shape of the target satellite, while maintaining sufficient details. Figure 2 illustrates various resolutions of the target satellite obtained with 196 elementary bodies (left), 14 bodies (centre), and 3 bodies (right).

Given the coordinate systems previously defined, the outer boundaries of the target satellite can be determined by using boundary points from the basic spherical and cylindrical shapes to trace out enveloping regions. Six points are used to define the outer boundary for the sphere and eight points for the cylinder as shown in Fig. 3. As the target satellite translates and rotates in space about its centre of mass, the outer boundary points trace out a *boundary envelop*, referred to as the *total box* (TB). The size of the TB is dependent on the motion of the satellite and possible intersection time between the servicer and the client spacecraft.

The TB determination process begins by using initial and final time to transit between the boundary constraints while plotting a straight line translation by the servicer drone from the beginning to the finishing point. The intersection between the straight drone path and the TB is found. From the intersection time, the guidance law reiterates the target satellite motion and replots the TB. TB is reduced in size as a result from the shortened intersection time. In other words, the idea consists of finding a intersecting line with the TB, compute the time of transit, and recompute the size of the TB. The iteration process continues until some tolerance on the intersecting time is reached. This process is typically limited to converges after a few iterations for practical implementation purposes. An illustration of TB resulting from a three-body target model rotating about two different axes of rotation is shown in Fig. 4.

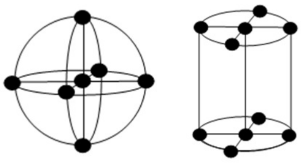

Figure 3. Elemental geometry boundary points.

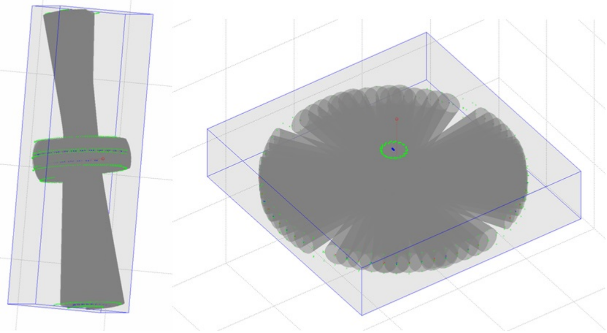

Figure 4. Total boxes traced by boundary points elementary geometry boundary points.

The timing for the intersection between the inspector paths with the TB is used to build the total number of constraint bodies to be used as constraints (i.e., geometrical obstacles) for the guidance algorithm. This will ultimately limit the final path by the servicer drone so it avoids most possible collisions. Spatial margin is added to the geometrical constrains for differences between the initial straight path and final determined path.

3.2. Legendre Polynomial Parameterization

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*. [2] The key feature of the ASTRO algorithm is its ability to optimize a trajectory-related cost function by projecting 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 constraints and path-length penalty functions can ensure a positive-definite cost function with respect to higher dimensional space of parameter errors, which, in most cases, guarantees asymptotic convergence directly to an admissible and optimal solution.

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 servicer drone spacecraft, respectively, and $\mathbf{x}_f(t_f)$ and $\dot{\mathbf{x}}_f(t_f)$ are the final position and velocity. The geometric 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. The inner loop of the algorithm satisfies Eqs. (1) and (4), 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 is the cost function that minimizes the path length, which is given by

$$S = \int_{x(t_0)}^{x(t_f)} \mathrm{d}s = \int_{t_0}^{t_f} \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2} \mathrm{d}t \tag{4}$$

where 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.

Equation (3) can be simplified by parametrizing 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 parametrized as

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

where i is the individual axis and C_{ik} are coefficients of the Legendre polynomials of order k that defines the drone spacecraft trajectory. Finally, 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

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

Since P_k' are derivatives of the standard Legendre polynomial, 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

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

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

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

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

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

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

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

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, 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 $\delta C_{ik} = 0$.

3.3. 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

$$\mathbf{X}_{BC} = \mathbf{P}_{BC}\mathbf{C} \quad (15)$$

or

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

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

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

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

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

where

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

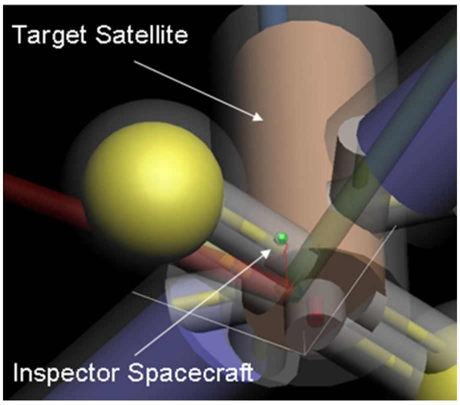

Figure 5. Safety buffer surrounding target satellite.

4. SIMULATION RESULTS

The guidance algorithm was developed in the MATLAB environment, the program 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 is 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 initially developed for ISS SPHERES testbed operations, was revised for this work so it can be combined with the front-end satellite dynamics motion propagation.

The satellite geometries used in simulations are provided in Table 1.

Table 1. Servicer drone and target satellite geometric properties

Description	Value
Servicer drone diameter	0.125 m
Target satellite height	31.27 m
Target satellite width	8.64 m
Target satellite depth	6.11 m

In the guidance law, the servicer spacecraft is modelled as a point mass, which requires the target satellite geometry to incorporate the dimensions of the servicer. As shown in Fig. 5, the inspector geometry is therefore accounted for by adding a safety buffer of the servicer radius to the target satellite, a 100% safety margin of servicer diameter is also added to the buffer for additional safety.

The boundary conditions of the servicer drone are defined as

$$\mathbf{x}(t_0) = [\ 0 \quad 0 \quad -10\ ]\ \mathrm{m}$$
$$\dot{\mathbf{x}}(t_0) = [\ 0 \quad 0 \quad 0\ ]\ \mathrm{m/s}$$

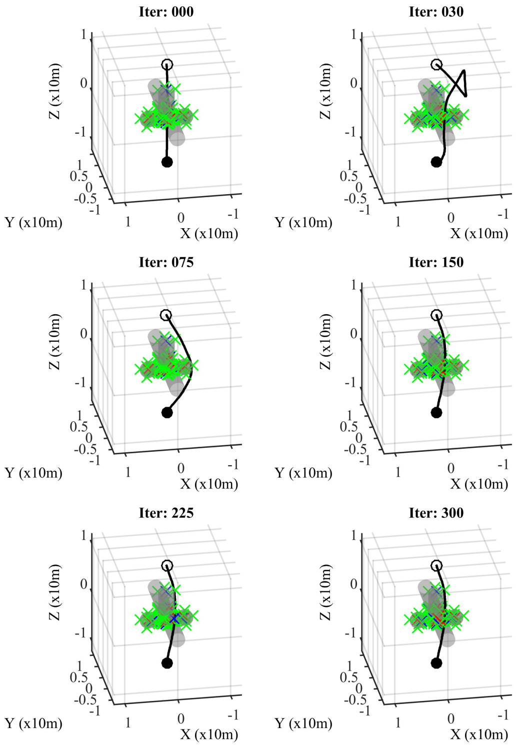

Figure 6. Optimal path around a stationary geostationary satellite.

$$
\begin{aligned}
\mathbf{x}(t_f) &= [\begin{array}{ccc} 0 & 0 & 10 \end{array}]\,\mathrm{m} \\
\dot{\mathbf{x}}(t_f) &= [\begin{array}{ccc} 0 & 0 & 0 \end{array}]\,\mathrm{m/s}
\end{aligned}
$$

Case 1: Stationnary Target. Simulation results illustrating the optimal solution around a stationary geostationary satellite is provided in Fig. 6. This simulation demonstrates the core ASTRO algorithm's capability in determining the optimal path after 155 outer-loop iterations. The total computation time for this scenario is 221 sec. It should be noted for this simulation, the 14 body fidelity model was used in the determination of the optimal trajectory.

Case 2: Tumbling Target. The target satellite motion is based on tumbling rotational rate of the GOES 8 geostationary satellite, as observed by Cognion *et al.* [16], i.e., 2.2, 21.8, and 2.2 deg/s about the roll, pitch, and yaw axis. Furthermore, it is assumed that the target satellite is travelling with a constant linear velocity of -0.01 m/s along the x-axis. Simulation results for number of iteration before reaching a feasible solution around a motioning geostationary satellite is provided in Fig. 7. Based on the feasibility output, a sub-optimal solution can be selected after 15 outer-loop iterations; the optimal solution however was not reached until after 31 iterations of outer-loop computation. The lower fidelity three body model was used for this simulation in order to balance the high number of total bodies to be considered for motion.

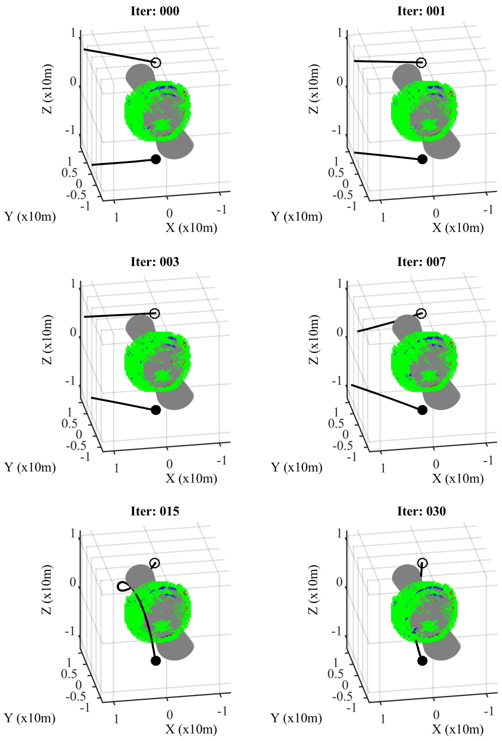

Figure 7. Optimal path around a motioning geostationary satellite.

The total time for reaching the optimal 31 iteration is 698 seconds; this is over three times the stationary simulation result. If a sub-optimal solution is forced by the user, the total computation time is reduced to 412 seconds. 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 are redundant with each other.

5. 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. Optimization techniques were developed to reduce computation time during real-time mission scenarios.

Future work will focus on limiting the actual feasibility points by introducing a limit corridor; this would eliminate wildly far reaching solutions that are not feasible in practice. Secondly, by introducing replacement geometry for the redundant full motion group will significantly reduce algorithm computational time. Finally, the total box (the region of high risk for collision occurrence) can be further reduced by introducing performance constraints such as acceleration limit and increased velocity when passing through it.

ACKNOWLEDGMENTS

This research was funded by the Natural Sciences and Research Council of Canada's Engage Grant program under the award EGP #469958-14. The authors also acknowledge Gregory E. Chamitoff, Alvar Saenz-Otero, Jacob G. Katz and David Miller from the MIT Space System Laboratory for their initial development of the ASTRO algorithm.

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] 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, 13-17 Jan 2014; also AIAA paper 2014-1290.

[3] 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.

[4] Munoz, J., Fitz-Coy, N., "Rapid Path-Planning Options for Autonomous Proximity Operations of Spacecraft," *AIAA Guidance, Navigation, and Control Conference*, Toronto, Ontario, August 2010; also Paper AIAA-2010-7667.

[5] Ulybyshev, Y., "Trajectory Optimization for Spacecraft Proximity Operations with Constraints," *AIAA Guidance, Navigation, and Control Conference*, Portland, Oregon, August 2011; also Paper AIAA-2011-6629.

[6] Hadaegh, F.Y., Kim, Y., and Mesbahi, M., "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.

[7] Hadaegh, F.Y., and Singh, G., "Collision Avoidance Guidance for Formation-Flying Applications," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Montreal, Canada, August 6-9, 2001; also Paper AIAA-2001-4088.

[8] 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.

[9] 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.

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

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

[12] Ranieri, C., "Path-Constrained Trajectory Optimization for Proximity Operations," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii, August 2008; also Paper AIAA-2008-6275.

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

[14] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge Univ. Press, Cambridge, England, U.K., 2004, Chaps. 2-5, 9-11.

[15] Lu, P., 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.

[16] Cognion, R., Albuja, A. A., and Scheeres, D. J., "Tumbling Rates of Inactive Geo Satellites," *International Astronautical Conference*, Toronto, Ontario, 29 Sep-3 Oct 2014; also IAC paper 14.C1.2.12x25119.