Sampling-Based Trajectory Generation for Autonomous Spacecraft Rendezvous and Docking

Griffin D. Francis[*], Emmanuel G. Collins, Jr.[†], Oscar Chuy[‡], and Aneesh Sharma[§]

Center for Intelligent Systems, Control, and Robotics (CISCOR)

Florida State University, Tallahassee, Florida, 32310, USA

This paper presents a methodology for computationally efficient, direct trajectory generation for autonomous spacecraft rendezvous using sampling with a distance or time cost function to be minimized. The approach utilizes a randomized A* algorithm called Sampling-Based Model Predictive Optimization (SBMPO) that exclusively samples the input space and integrates the dynamic model of the system. A primary contribution of this paper is the development of appropriate optimistic A* heuristics that take into account a goal position and velocity and are based on the minimum distance and minimum time control problems for the vehicle. These heuristics enable fast computation of trajectories that end in zero relative velocity. Additionally, this paper introduces an alternative approach to collision avoidance for use in graph-based planning algorithms. By referencing the full state of the vehicle, a method of imminent collision detection is applied within the graph search process of the trajectory planner. Using a six degree-of-freedom relative motion spacecraft dynamic model, simulation results are illustrated for generating rendezvous-feasible trajectories in cluttered environments.

Nomenclature

r	Position vector, m
v	Velocity vector, m/s
a	Acceleration vector, m/s^2
ω	Angular velocity vector, deg/s
q	Rotation quaternion vector
Θ	Rotation matrix
Ω	Kinematic quaternion matrix
m	Mass, kg
J	Inertia matrix
u	Thrust control input vector
τ	Torque control input vector

I. Introduction

As indicated by recent NASA studies and the 2010 US Space Policy, there is an imminent need to develop technologies for the mitigation of orbital space debris.[1] It has been suggested that it will be necessary to conduct a minimum of five debris removal missions per year beginning in 2015 in order to merely maintain the current population of space debris.[2] The increasingly cluttered state of orbital space poses a threat to the future success of manned and unmanned space flight.

[*]Graduate Research Assistant, Department of Mechanical Engineering.

[†]Professor, Department of Mechanical Engineering.

[‡]Research Scientist, Department of Mechanical Engineering.

[§]Graduate Research Assistant, Department of Computer Science.

A major hurdle in achieving such technologies is the development of adequate trajectory planning methods. Intelligent autonomous navigation requires the generation of paths and control tracking inputs to traverse a route towards a goal. This intelligence is achieved via a planning algorithm that is, ideally, capable of quickly determining an optimal path and control inputs subject to defined system constraints.

Partly motivated by the necessity to mitigate the threats associated with the growing presence of space debris, various autonomous spacecraft missions have been undertaken, including Orbital Express and Spacecraft for the Universal Modification of Orbits/Front-end Robotics Enabling Near-term Demonstration (SUMO/FREND) by DARPA and Autonomous Nanosatellite Guardian for Evaluating Local Space (ANGELS) by Air Force Research Laboratory (AFRL). Future advancement of autonomous spacecraft navigation requires the development of innovative planning algorithms capable of generating trajectories in orbit.

With varying degrees of success, several methods have been attempted for solving the spacecraft trajectory planning problem. These methods are largely representative of the planning techniques already employed for robotic manipulators and wheeled vehicles. Spanning the various algorithms used for ground-based robots, mixed integer linear programming (MILP),[3,4] mixed integer nonlinear programming (MINLP),[5] potential functions,[6] rapidly exploring random trees (RRTs),[7,8] and calculus of variations[9] have been researched as possible solutions.

Critically necessary for aerospace work, the implementation of the system dynamic model allows for explicit consideration of propulsion constraints when planning control inputs. Many of these algorithms are hampered by an inability to deal with the nonlinear dynamics of the spacecraft and/or a lack of combined position and orientation planning capabilities. Furthermore, several are incapable of planning in the presence of obstacles. Only a few report computation times and these times typically prohibit the possibility of using the methods in real-time.

Within the realm of the more general topic of direct trajectory generation, recent research has led to approaches that employ optimization and randomly-generated graphs.[10,11,12] These methodologies seek to take advantage of the efficiency obtained via algorithmic optimization and the robustness of randomized, sampling-based graph formation. When implemented with A*-type optimization, random graph methods can incorporate a heuristic function that facilitates rapid computation of optimal trajectories. Unlike the randomized A* approach,[11,12] methods such as RRT*[10] and sampling-based RRT[13] do not utilize A* optimization and, as a result, do not benefit from the efficiency achieved by using the prediction associated with an optimistic heuristic.

Sampling Based Model Predictive Optimization (SBMPO)[11,12] incorporates sampling and A* optimization to generate trajectories. SBMPO has been demonstrated as an effective and efficient trajectory planning technique for autonomous underwater vehicles (AUVs),[14] ground-based mobile robots,[15,16,17] and robotic manipulators.[18]

Motivated by the promising results obtained using SBMPO in navigation and planning tasks for ground-based robots, ongoing SBMPO research is focused on the continued refinement of the algorithm and application to additional autonomous systems. In this work, the SBMPO algorithm is studied for application to autonomous spacecraft rendezvous and docking duties with the ultimate goal of real-time trajectory planning for orbital vehicles.

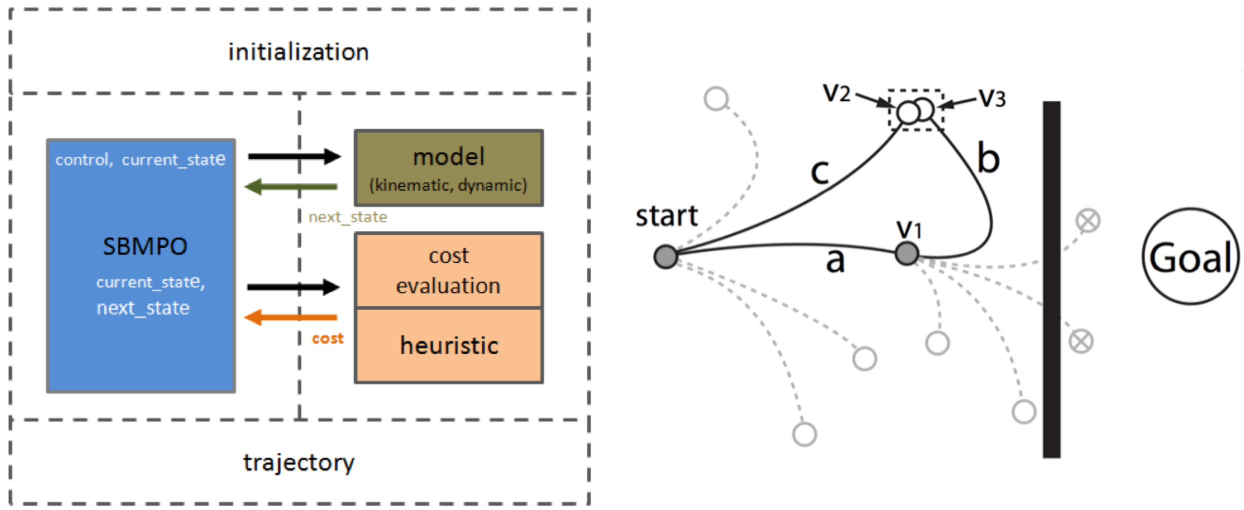
The efficiency of SBMPO is closely linked to the development of an appropriate optimistic A* heuristic. In this paper, two recently developed heuristic functions, that are based on the solutions to the minimum time[18] and minimum distance[19] control problems, are presented. These heuristics facilitate the rapid computation of trajectories that terminate with zero relative velocity, which is a requirement for spacecraft rendezvous trajectory planning. Potentially useful in other approaches toward randomized A* planning, a primary contribution of this paper is the application of these heuristics to the generation of rendezvous trajectories.

II. Preliminaries

This section provides a brief description of SBMPO and identifies the details of the spacecraft rendezvous problem within the context of trajectory generation.

A. Brief Overview of Sampling-Based Model Predictive Optimization

SBMPO is a sampling-based algorithm that can be used for motion planning with kinematic and dynamic models. It can plan using a variety of cost functions, including the standard sum of the squared error cost function that is commonly used in Model Predictive Control (MPC).[20] At its inception, SBMPO was motivated by a desire to utilize sampling and A*-type optimization in lieu of the nonlinear programming strategies that are commonly employed for optimization in MPC. Use of these techniques provides SBMPO with the ability to avoid local minima, when present, and maintain fast computations when implemented with properly designed A* heuristics.



(a) (b)

Figure 1. (a) Block diagram describing SBMPO trajectory planning. (b) Illustration of the graph expansion process in SBMPO.

The block diagram, shown in figure 1(a), identifies the essential structure of the SBMPO algorithm when used for trajectory generation. The model, cost evaluation, and heuristic are supplied by the user and may be tailored to suit certain aspects that are characteristic of the specific planning scenario. To enable the trajectory generation process within the SBMPO algorithm, a graph is created in the planning space from the start to the goal that consists of connected vertices that identify the states of the system, the control inputs, and the cost associated with the states. Furthermore, to avoid infinitesimal improvements as vertices are expanded, an implicit grid is imposed on the graph to discretize the planning space. A simplified visualization of the graph generation process is shown in figure 1(b), which illustrates the discretization of the planning space by identifying vertices V2 and V3 as occupying the same cell of the implicit grid. The following are the main steps of SBMPO:[11,12]

1. **Select a vertex with highest priority in the queue:** The vertices are collected in the *open list*, which ranks the potential expansion by their priority. The open list is implemented as a heap so that the highest priority or lowest cost vertex that has not been expanded is on top. If the selected vertex is the goal, SBMPO terminates. If not, the algorithm goes to step 2. The vertex representing the start will have the highest priority as an initialization.

2. **Sample input space:** Generate a sample of the inputs to the system that satisfies the input constraints. The input sample and current state are passed to the system model and the system is integrated to determine the next state of the system. It should be noted that the current state is the state of the selected vertex.

3. **Add new vertex to the graph:** Check if the graph already contains a vertex whose state maps to the new state of the system. If the vertex exists, only add an edge from the current vertex (i.e., the selected vertex) to the vertex whose state is the same as the new state. Otherwise, add a vertex whose state is the next state.

4. **Evaluate new vertex cost:** Use an A* heuristic to evaluate the cost of the generated vertices based on the desired objective (i.e., shortest distance, shortest time, or least amount of energy, etc.). Add a new vertex to the priority queue based on the vertexs cost.

5. **Repeat 2-4:** Repeat steps 2-4 for a user-defined number of successors.
6. **Repeat 1-5 until one of the stopping criteria is true:** Steps 1-5 are repeated until the goal is found or the maximum number of allowable iterations is achieved.

Table 1. Planner Performance Comparison

	SBMPO	RRT*
Distance (m)	7.39	8.28
Comp. Time (ms)	1.9	50.0

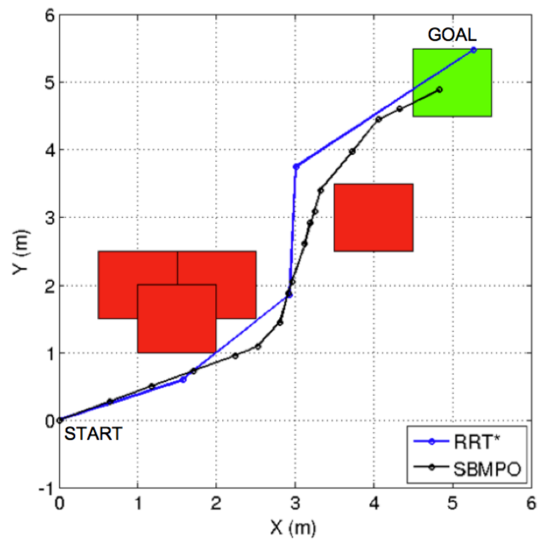


Figure 2. Comparison of SBMPO and RRT*.

By virtue of this strategy, SBMPO offers superior performance when compared with other planning algorithms. For comparison, consider a typical planning scenario for a mobile robot with a kinematic model given by

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} v_t \cos\theta_{k+1} \\ v_t \sin\theta_{k+1} \\ v_\theta \end{bmatrix} \Delta T, \qquad (1)$$

where x and y are the vehicle position components, θ is the vehicle heading, and the control inputs v_t and v_θ are the vehicle's forward velocity and rotational velocity, respectively. As indicated by the planning results shown in figure 2 and summarized in table 1, SBMPO generates trajectories that are similar to those formed with RRT*, but performs the calculation more than one order of magnitude faster. In complicated planning scenarios, this significant discrepancy in computation time prohibits the use of RRT* and similar approaches. Evident in the simple planning scenario shown in this comparison, the use of a heuristic for predicition facilitates the rapid computation in SBMPO.

B. Six-DOF Relative Motion Spacecraft Model for Rendezvous

The spacecraft rendezvous and docking work presented in this paper requires the incorporation of the vehicle dynamic model for full six degree of freedom (DOF) trajectory planning. Shown in control affine form, the nonlinear dynamic equation describing the relative motion of the spacecraft with respect to the target is

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\omega} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \tilde{0} \\ \dot{v} \\ -J^{-1}\omega \times J\omega \\ \frac{1}{2}\Omega(\omega)q \end{bmatrix} + \begin{bmatrix} \frac{1}{m}\Theta^T(q)u(t) \\ \tilde{0} \\ J^{-1}\tau(t) \\ \tilde{0} \end{bmatrix}, \qquad (2)$$

where r and v represent the position and velocity of the spacecraft with respect to the target, ω is the angular velocity in the spacecraft body frame, and $\tilde{0}$ is the zero vector. As in McCamish,[21] a spacecraft with the characteristics defined in table 2 was used in simulation.

Essentially, the rendezvous and docking task may be simplified as a trajectory generation problem for an autonomous chase spacecraft navigating relative to a target orbiting body. A successful solution to the problem requires the determination of the path and control inputs necessary to achieve motion from an initial position and orientation to some goal position and orientation. During the planning from initial to goal configuration, the algorithm considers obstacles and other spatial restrictions, known motion and/or acceleration of the target, and, in many cases, nonholonomic vehicle constraints. Also, unlike path generation, the trajectory generation problem involves planning with conditions placed on the goal ending velocity.

III. Combined Relative Position and Attitude Trajectory Planning Using a Dynamic Model

Unlike typical mobile robotics applications of A*-type path planning, spacecraft rendezvous trajectory planning requires additional consideration of motion constraints that are fundamentally dictated by the nature of spacecraft docking. Specifically, the docking task requires that the spacecraft approach the goal in a manner that seeks to match the motion of the target and prevents unwanted collision.

SBMPO enables the development of trajectories based on optimizing various physical metrics (e.g., distance, time, or energy). This versatility depends on the development of an appropriate heuristic used in the A*-type planner.

In order to facilitate rapid computation of trajectories, SBMPO relies on a heuristic that predicts the cost-to-goal as nodes are explored by the planner. Enabling the algorithm to efficiently converge to an optimal solution, the A*-type planner requires an optimistic heuristic that is fairly non-conservative. When implemented with a naïve or overly-conservative heuristic, A*-type algorithms are very computationally inefficient. Therefore, it is crucial that the heuristic be carefully developed within the context of the planning scenario.

A. Development of an Appropriate Minimum Time Heuristic

The necessity of forming a well-developed heuristic is evident when considering the problem of time-optimized trajectory planning for a rendezvous scenario.

First, consider the equation describing the motion of a particle given by

$$r = v_i t + \frac{1}{2} a_i t^2, \tag{3}$$

where r is the distance to the goal, v_i is the current velocity, and a_i is the current acceleration. Based on this relationship, the predicted minimum time to reach the goal would be expressed by

$$t = \frac{-v_i \pm \sqrt{v_i^2 - 2a_i r}}{a_i}. \tag{4}$$

Although a minimum time heuristic formed using this relationship is optimistic, the algorithm fails to efficiently converge to a solution when it is implemented. When applied to simple planning scenarios similar that shown in figure 2, this naïve minimum time heuristic fails to produce a solution in a reasonable number of iterations (i.e., fewer than one million).

To enable efficient minimum time planning, an appropriate heuristic was formed using the solution to the fundamental time optimal control problem. The minimum time control problem can be solved by forming the Hamiltonian and applying Pontryagin's Maximum Principle.[22] Assuming that the controlled acceleration is bound by

$$-\underline{a} \leq a \leq \bar{a}, \tag{5}$$

where $\underline{a}$ and $\bar{a}$ are, respectively, the lower and upper limits for acceleration. The solution of the minimum time control problem of Bryson[22] can be generalized to yield

$$\begin{aligned} t^2 - \frac{2v_i}{\underline{a}} t &= \frac{v_i^2 + 2(\underline{a} + \bar{a})r}{\underline{a}\bar{a}}, \text{ if } r + \frac{v_i |v_i|}{2\bar{a}} < 0 \\ t^2 + \frac{2v_i}{\bar{a}} t &= \frac{v_i^2 - 2(\underline{a} + \bar{a})r}{\underline{a}\bar{a}}, \text{ if } r + \frac{v_i |v_i|}{2\underline{a}} > 0. \end{aligned} \tag{6}$$

The minimum time computed using Eq. (6) corresponds to the bang-bang optimal controller shown in figure 3, which illustrates switching curves that take the system to the origin by applying either the minimum or maximum control inputs (i.e., $u = -\underline{a}$ or $u = \bar{a}$). Depending on the intial conditions, the system uses either the minimum or maximum control input to take the system to the appropriate switching curve. For example, if (r_i, v_i) corresponds to point p_1 in figure 3, then the control input $u = -\underline{a}$ is applied until the system reaches point p_2 on the switching curve. At p_2, the control input is then switched to $u = \bar{a}$, which drives the system to the goal where $r = v = 0$.

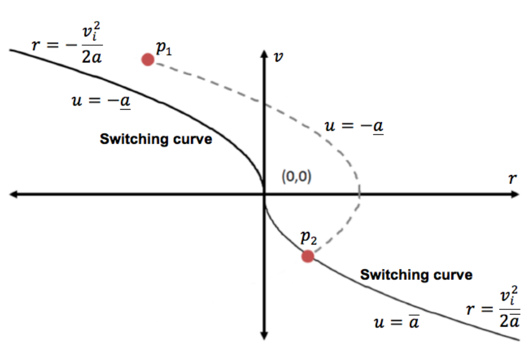


Figure 3. Minimum time control curve.

B. Development of an Appropriate Minimum Distance Heuristic

For minimum distance *path-only* planning, it is sufficient to use a heuristic that represents the Euclidean distance between the expanded vertex and the goal. However, when applied in a planner that propagates the vehicle's trajectory using the dynamic model, this heuristic leads to a minimum distance trajectory in which the vehicle accelerates, rather than brakes, as it approaches the goal.

In scenarios such as autonomous rendezvous and docking, where path-only planning is inappropriate, it is necessary to develop a heuristic that considers the full state of the vehicle throughout the path expansion process as the planner develops a trajectory that seeks to match the goal state. For minimum distance trajectory planning, the heuristic that was developed is based on the fact that the distance to goal r, the initial velocity v_i, the final velocity v_f, and the acceleration a are related by

$$v_f^2 = v_i^2 + 2ar. \tag{7}$$

In order to stop at the goal, $v_f = 0$, and the relationship becomes

$$r_{stop} = -\frac{v^2}{2\underline{a}}, \tag{8}$$

where r_{stop} is the minimum stopping distance, v is the velocity, and the sampled control thrust u is bounded by

$$-m\underline{a} \leq u \leq m\bar{a}. \tag{9}$$

Algorithm 1 minDistHeuristic(r, v, r_G)

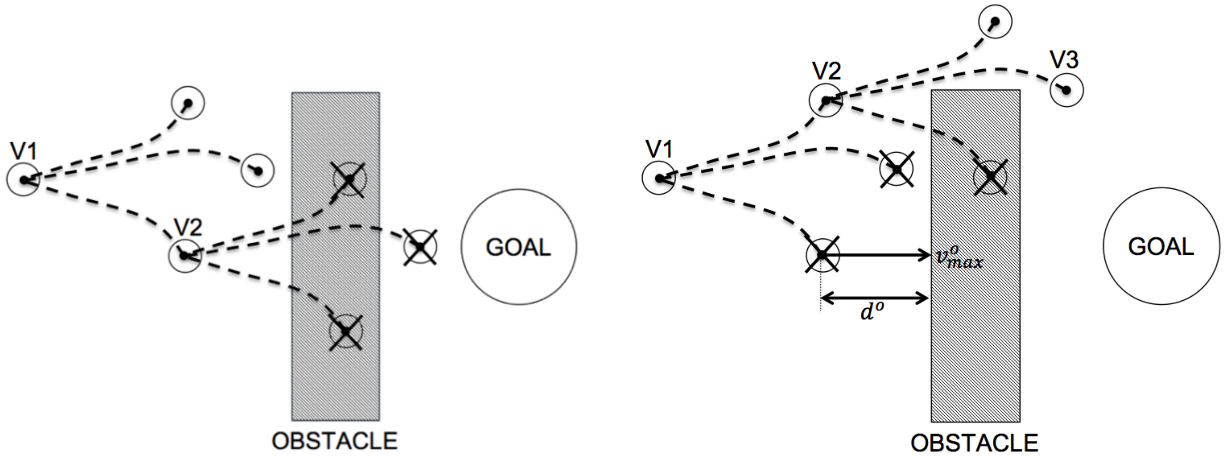
1: **if** $r < r_G$ **then**
2: $\quad r_{goal} = r_G - r$
3: $\quad$ **if** $v < 0$ **then**
4: $\quad\quad r_{stop} = \frac{-v^2}{\bar{a}}$
5: $\quad\quad$ **return** $2r_{stop} + r_{goal}$
6: $\quad$ **else if** $v > 0$ **then**
7: $\quad\quad r_{stop} = \frac{-v^2}{\underline{a}}$
8: $\quad\quad$ **if** $r_{stop} > r_{goal}$ **then**
9: $\quad\quad\quad$ **return** $2r_{stop} - r_{goal}$
10: $\quad\quad$ **else**
11: $\quad\quad\quad$ **return** r_{goal}
12: $\quad\quad$ **end if**
13: $\quad$ **else**
14: $\quad\quad$ **return** r_{goal}
15: $\quad$ **end if**
16: **else if** $r > rG$ **then**
17: $\quad r_{goal} = r - r_G$
18: $\quad$ **if** $v < 0$ **then**
19: $\quad\quad r_{stop} = \frac{-v^2}{\bar{a}}$
20: $\quad\quad$ **if** $r_{stop} > r_{goal}$ **then**
21: $\quad\quad\quad$ **return** $2r_{stop} - r_{goal}$
22: $\quad\quad$ **else**
23: $\quad\quad\quad$ **return** r_{goal}
24: $\quad\quad$ **end if**
25: $\quad$ **else if** $v > 0$ **then**
26: $\quad\quad r_{stop} = \frac{-v^2}{\underline{a}}$
27: $\quad\quad$ **return** $2r_{stop} + r_{goal}$
28: $\quad$ **else**
29: $\quad\quad$ **return** r_{goal}
30: $\quad$ **end if**
31: **else**
32: $\quad$ **if** $v < 0$ **then**
33: $\quad\quad r_{stop} = \frac{-v^2}{\bar{a}}$
34: $\quad\quad$ **return** $2r_{stop}$
35: $\quad$ **else if** $v > 0$ **then**
36: $\quad\quad r_{stop} = \frac{-v^2}{\underline{a}}$
37: $\quad\quad$ **return** $2r_{stop}$
38: $\quad$ **else**
39: $\quad\quad$ **return** 0
40: $\quad$ **end if**
41: **end if**

The complete heuristic, given by algorithm 1, has three input parameters: the relative position of the vehicle r, the relative velocity of the vehicle v, and the desired relative position of the vehicle r_G. The algorithm returns r_{stop}, the minimum stopping distance required to reach the goal at a desired velocity subject to the acceleration bounds defined within the sampler. In the event that the vehicle is approaching the goal too fast and will not be able to stop, lines 1-15 of the algorithm return a minimum stopping distance that requires the vehicle to go past the goal and return. Lines 16-30 deal with the mirror case in which the vehicle has surpassed the goal. In the special case where the vehicle is at the goal but the velocity is not zero, lines 31-41 are referenced. In the event that that vehicle is at the goal and has a zero velocity, lines 30-40 are used and the heuristic is returned as 0.

C. Development of Momentum-Aware, Imminent Collision Detection

When planning in cluttered environments, the algorithm requires a collision detection method to reject vertices that spatially violate obstacles. Typically, obstacle avoidance is achieved by simply eliminating vertices that result in direct collision. This method of basic collision detection is illustrated in figure 4(a).

For full-state trajectory planning, rudimentary collision detection does not consider additional constraints placed on spatially viable vertices in the vicinity of obstacles that are none-the-less invalid for further



(a) Nodes rejected as collision is detected.

(b) Nodes preemptively rejected.

Figure 4. Illustrations of (a) rudimentary collision detection and (b) imminent collision detection.

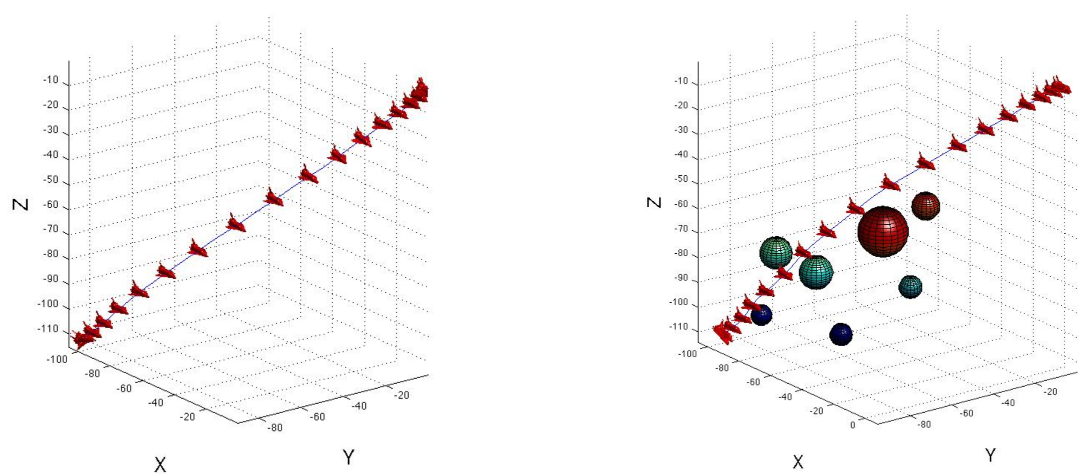
expansion. For example, suppose that a vertex is generated that, based on its position, does not collide with an obstacle but due to the velocity required to achieve the state at that vertex will, when expanded, inevitably only generate child vertices that are in collision with an obstacle. Eventually, the vertex described would be ignored, but at the cost of computational performance. Ideally, such a vertex, like the one labeled V2 in figure 4(a), would be preemptively rejected for expansion.

To improve performance, a momentum-based imminent collision detection method was developed for this problem. Using the same relationship applied to the minimum distance heuristic, imminent collision detection is based on the vehicle's ability to decelerate prior to collision. When used for collision detection, the relationship is

$$v^o_{max} = \sqrt{2\underline{a}d^o}, \tag{10}$$

where v^o_{max} is the maximum viable velocity in the direction of the nearest obstacle and d^o is the minimum distance to the nearest obstacle. Using imminent collision detection, the vertices are expanded and rejected as shown in figure 4(b).

IV. Simulation Results for Combined Relative Position and Attitude Planning



(a) Minimum distance trajectory.

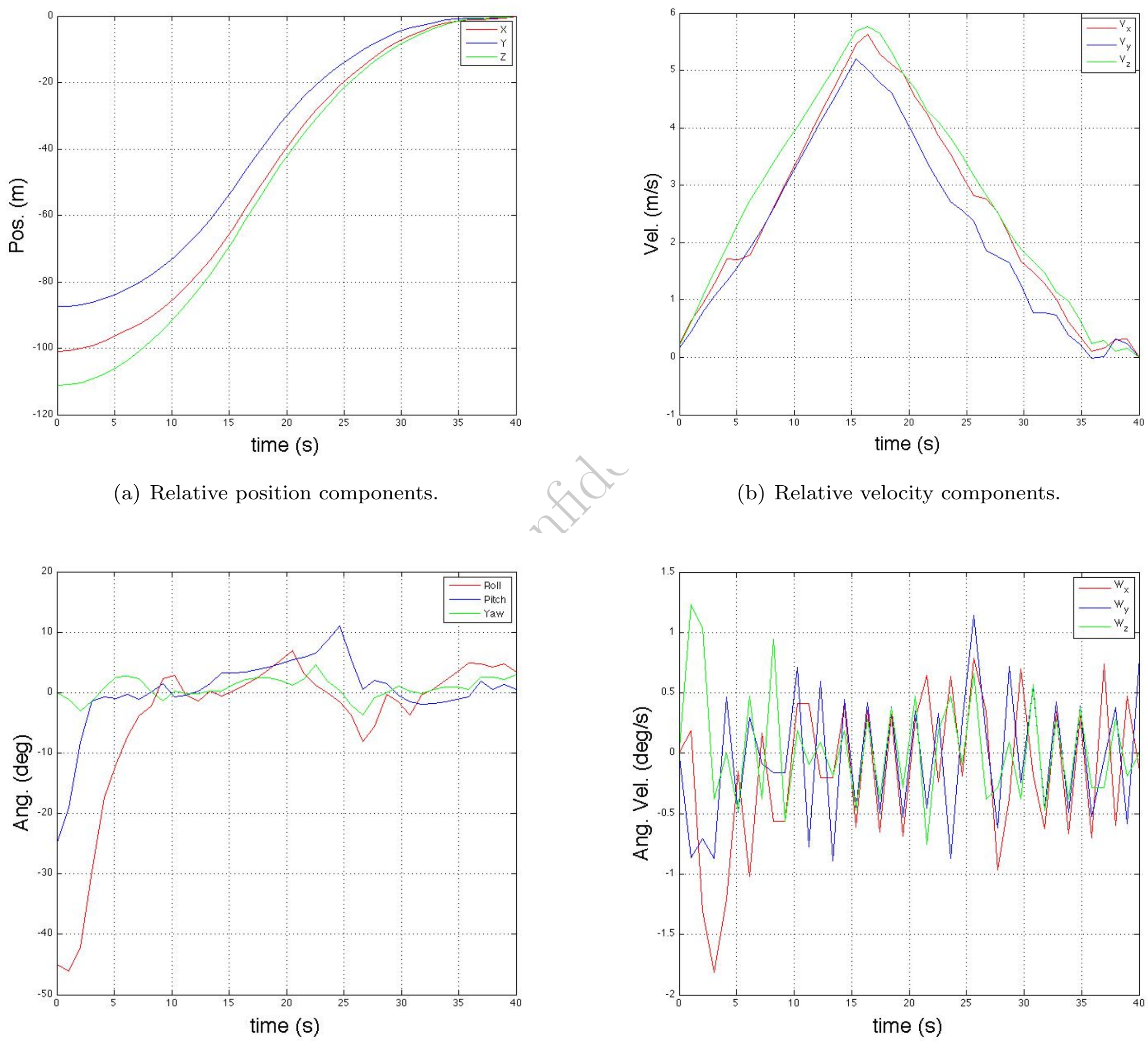
(b) Minimum time trajectory.

Figure 5. Planned trajectories in uncluttered and cluttered space.

The results presented in this section demonstrate the capability of the planner to rapidly generate six-DOF trajectories that are appropriate for rendezvous in uncluttered and cluttered environments. These results utilize the heuristics that are defined in Sections III-A and III-B to minimize time and distance, respectively. When implemented with more conservative heuristic functions, SBMPO failed to converge to a solution trajectory within the maximum number of iterations allowed.

In each of the simulations, the spacecraft is initially misoriented with Euler angles of $(-45.0°, -25.0°, 0.0°)$ and is positioned at $(-101.0\text{m}, -87.5\text{m}, -111.2\text{m})$. In this configuration, the spacecraft is trailing the target, which, as defined by the system dynamics, is located at the frame origin. Intending to rendezvous with the target, the goal position and orientation is coincident with the frame origin.

A. Simulation Results in Uncluttered Environment Using Minimum Distance Heuristic



(a) Relative position components.

(b) Relative velocity components.

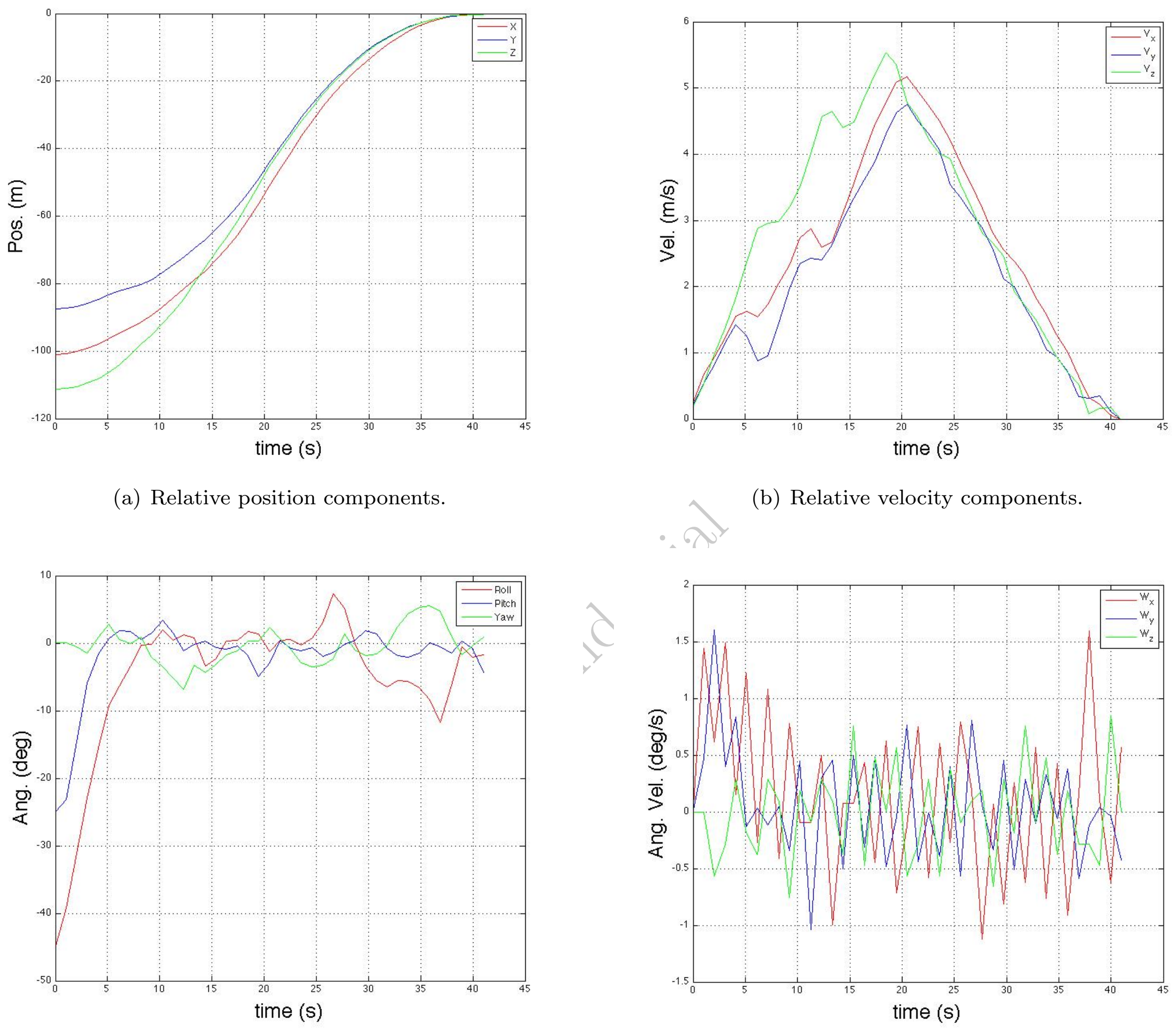
(c) Euler angle components.

(d) Angular velocity components.

Figure 6. Parameter profiles for minimum distance trajectory.

Most evident when deployed in obstacle-free environments, the planner generates minimum distance trajectories that result in a desired relative position, attitude, and velocity with respect to the target. The results in this section primarily serve to illustrate the effectiveness of the developed heuristic when used in conjunction with the SBMPO algorithm to generate rendezvous feasible trajectories. Shown in figures 5(a) and 6, simulation results for uncluttered planning correspond to an expected control sequence, where the vehicle decelerates as it closes in on the goal. When planning in uncluttered space, the minimum distance solution approximately matches the minimum time solution driven by the bang-bang control sequence and is optimal subject to randomized sampling. The rendezvous trajectory, corresponding to a path length of 174.02m, was generated in 0.483s.

B. Simulation Results in Cluttered Environment Using Minimum Time Heuristic



(a) Relative position components.

(b) Relative velocity components.

(c) Euler angle components.

(d) Angular velocity components.

Figure 7. Parameter profiles for minimum time trajectory.

SBMPO is also capable of generating time-optimal, collision-free trajectories when in cluttered environments, which is desirable for the potential application of the algorithm to orbital debris navigation and removal. Shown in figures 5(b) and 7, simulation results of minimum time planning in cluttered environments demonstrate the capability of SBMPO to generate feasible rendezvous trajectories in the presence of obstacles. In this simulation, seven randomly-sized and randomly-positioned spherical objects were included as obstacles impeding the path toward the target. As predicted, the resultant minimum time trajectory corresponds to the optimal bang-bang control sequence with some expected divergence due to the presence of impeding obstacles. The collision-free rendezvous trajectory, corresponding to a path length of 179.23m, was generated in 2.233s.

V. Conclusion

This paper has presented a sampling-based approach to six-DOF spacecraft rendezvous trajectory planning for orbital debris mitigation. The trajectory planning was achieved via Sampling-Based Model Predictive Optimization (SBMPO), a sampling-based algorithm that operates within the system input space.

Facilitating the efficiency of the randomized A* algorithm, this paper presented the construction of two appropriate, optimistic A* heuristics that are based on the solutions to the minimum distance and minimum time control problems. These heuristics enable rapid computation of rendezvous trajectories that end in zero relative velocity.

Considering an obstacle-free planning environment, SBMPO was applied to the spacecraft relative motion model to determine a rendezvous-feasible, minimum distance trajectory. The presentation of the obstacle-free simulation in this paper served to demonstrate the effectiveness of the optimistic A* heuristic and as a comparative baseline.

A necessity when used in orbital debris mitigation missions, SBMPO was demonstrated to be capable of generating rendezvous trajectories in the presence of known obstacles. By implementing a well-developed minimum time heuristic, the algorithm rapidly converged to an optimal solution trajectory that accomodates the rendezvous scenario. Furthermore, the computational performance, despite the obstacles, was comparable to that of obstacle-free planning.

When planning in the presence of obstacles, the efficiency of the algorithm was, in part, due to the use of the imminent collision detection method that is highlighted in figure 4(b). Although not explicitly presented in the results, it should be noted that this collision detection method may be most effective when deployed in scenarios with a high population and/or high density of obstacles because the number of nodes explored is reduced when imminent collision detection is employed.

Based on the computational quickness of the algorithm, this approach is a likely candidate for use in real-time guidance and navigation. Lending to the viability of future autonomous orbital debris removal missions, the computationally rapid results shown in this paper indicate the promising potential for the deployment of SBMPO in autonomous rendezvous and docking problems.

Future work will focus on the implementation of rapid replanning in the algorithm, the identification of planning issues associated with onboard perception of obstacles, and the development of additional optimization metrics, such as minimum fuel consumption. Additionally, future study will investigate unique planning issues associated with spacecraft rendezvous and docking, such as thruster plume impingement and accommodating additional environmental constraints.

Appendix

Table 2. Spacecraft Characteristics

Parameter	Value
Length	1.0 m
Width	1.0 m
Height	1.0 m
Mass	100.0 kg
Moment of Inertia X	16.67 kg-m^2
Moment of Inertia Y	16.67 kg-m^2
Moment of Inertia Z	16.67 kg-m^2
Number of Thrusters	6
Maximum Thrust Per Axis	1.0 N
Number of Reaction Wheels	3
Rotation Wheel Maximum Torque	0.055 Nm

References

[1]Obama, B., *National space policy of the United States of America*, Executive Office of the President, 2010. [Online] http://books.google.com/books?id=lNTDYgEACAAJ

[2]Liou, J., and Johnson, N.,J. Liou and N. Johnson, "A sensitivity study of the effectiveness of active debris removal in leo," *Acta Astronautica*, vol. 64, no. 2-3, pp. 236–243, 2009.

[3]Breger, L., and How, J., "Safe trajectories for autonomous rendezvous of spacecraft," *Journal of Guidance, Control and Dynamics*, vol. 31, no. 5, pp. 1478–1489, 2008.

[4]Richards, A., Schouwenaars, T., How, J., and Feron, E., "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 755–764, 2002.

[5]Zhang, J., Tang, G., Luo, Y., and Li, H., "Orbital rendezvous mission planning using mixed integer nonlinear programming," *Acta Astronautica*, 2010.

[6]Zhang, D., Song, S., and Pei, R., "Dynamic obstacle avoidance of autonomous rendezvous and docking using potential function guidance based-fuzzy logic system," in *IEEE International Symposium on Systems and Control in Aeronautics and Astronautics (ISSCAA)*, pp. 34–39, 2010.

[7]Wang, P., Chen, C., Guo, J., and Cui, N., "Randomization-based algorithms for real-time servicing spacecraft motion planning on elliptical orbit," in *IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 4924–4929, 2009.

[8]Aoude, G., How, J., and Garcia, I., "Two-stage path planning approach for designing multiple spacecraft reconfiguration maneuvers," in *International Symposium on Space Flight Dynamics*, 2007.

[9]Henshaw, C., and Sanner, R., "Variational technique for spacecraft trajectory planning," *Journal of Aerospace Engineering*, vol. 23, p. 147, 2010.

[10]Karaman, S., and Frazzoli, E., "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[11]Dunlap, D. D., Caldwell, C. V., Collins, E. G., and Chuy, O., "Motion planning for mobile robots via sampling-based model predictive optimization," in *Mobile Robots*, InTech, 2011.

[12]Dunlap, D. D., Caldwell, C. V., and Collins, E. G., "Nonlinear model predictive control using sampling and goal-directed optimization," in *IEEE International Conference on Control Applications (CCA)*, pp. 1349 –1356, 2010.

[13]LaValle, S., and Kuffner, J., "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[14]Caldwell, C. V., Dunlap, D. D., and Collins, E. G., "Motion planning for an autonomous underwater vehicle via sampling based model predictive control," in *OCEANS 2010*, pp. 1 –6, 2010.

[15]Ordonez, C., Gupta, N., Collins, E. G., Clark, J. E., and Johnson, A. M., "Power modeling of the XRL hexapedal robot and its application to energy efficient motion planning, in *Proceedings of the 15th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, Baltimore, MD, July 23-26, 2012.

[16]Ordonez, C., Gupta, N., Yu, W., Chuy, O., and Collins, E. G., "Modeling of Skid-Steered Wheeled Robotic Vehicles on Sloped Terrains, in *Proceedings of the ASME Dynamic Systems and Control Conference*, Fort Lauderdale, Florida, Oct. 17-19, 2012.

[17]Ordonez, C., Gupta, N., Chuy, O., and Collins, E. G., "Momentum Based Traversal of Mobility Challenges for Autonomous Ground Vehicles, in *IEEE Conference on Robotics and Automation*, May 2013, Munich, Germany.

[18]Chuy, O., Collins, E. G., Dunlap, D. D., and Sharma, A., "Sampling-Based Direct Trajectory Generation Using the Minimum Time Cost Function, in *13th International Symposium on Experimental Robotics*, Qubec, Canada, June 17-21, 2012.

[19]Mann, T., "Application of Sampling-Based Model Predictive Control to Motion Planning for Robotic Manipulators," M. S. Thesis, Florida State University, December 2010.

[20]Maciejowski, J., *Predictive Control with Constraints*, Pearson Education, 2002.

[21]McCamish, S., Romano, M., and Yun, X., "Autonomous distributed control of simultaneous multiple spacecraft proximity maneuvers," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 630–644, 2010.

[22]Bryson, A. E., and Ho, Y., *Applied Optimal Control: Optimization, Estimation, and Control*, Hemisphere Publishing Corporation, 1975.

[23]Fehse, W., *Automated rendezvous and docking of spacecraft*, Cambridge University Press, 2003.

[24]LaValle, S., *Planning Algorithms*, Cambridge University Press, 2006.

[25]Sidi, M., *Spacecraft Dynamics and Control: A Practical Engineering Approach*, Cambridge University Press, 2000.