In-Space Inspection Maneuver Analysis Using Trajectory Optimization

Ian Connerney[1]

Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24060

Alejandro Sosa[1], Kevin Schroeder[2], Jonathan Black[3]

Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24060

Abstract

The capability to perform in-space inspection and characterization of space objects is central to the next generation of space situational awareness. The ability to diagnose and respond to spacecraft anomalies is often hampered by the lack of capability to perform inspection or testing on the target vehicle in flight. While some limited ability to perform inspection can be provided by an extensible boom, such as the robotic arms deployed on the space shuttle and space station, a free-flying companion vehicle provides maximum flexibility of movement about the target. Safe and efficient utilization of a companion vehicle requires trajectories capable of minimizing spacecraft resources, e.g., time or fuel, while adhering to complex path and state constraints. This paper investigates and compares solution methods used to find optimal trajectories for a variety of potential inspection maneuvers subject to complex constraints. The two solution methods investigated are a randomness-based adapted A* Search method and a nonlinear optimization method based on direct collocation. We investigate trajectories utilizing impulsive burns and continuous thrusting, as well as problems including additional constraints, such as those with complex keep-out-regions and thruster plume limitations that might be required for inspection of a specific target area in a complex environment. This work is widely applicable and can be expanded to apply to a variety of relative trajectory problems. One such example involves with multiple inspector satellites working together for in-space inspection maneuvers in need of efficient computation of complicated relative motion trajectories.

1. Introduction

This introduction provides a basic understanding of autonomous spacecraft trajectory optimization, demonstrates its need as an enabling technology for future spacecraft missions, and describes challenges associated with trajectory design that motivate the need for advancement in trajectory planning methods.

Spacecraft trajectory planning refers to the process of generating a dynamically feasible state and control plan for the spacecraft to execute to accomplish a goal. In this paper, the process of *relative* satellite trajectory planning (satellite trajectory planning relative to another spacecraft or object in orbit) will be described in depth along with a comprehensive investigation into relevant solution methods with a focus on supporting *autonomous* relative guidance. Relative satellite trajectory planning can be thought of as the trajectory planning portion of a standard rendezvous and proximity operations (RPO) mission. An RPO mission is often described by the motion of one satellite, usually referred to as the "deputy" or "follower", with respect to the second, normally uncooperative satellite (or other space object), referred to as a "chief" or "target." In this research we consider only the final phases of an RPO mission, the "close range rendezvous operations" which by definition begins at a relative distance of a few kilometers to the chief [1].

Here *autonomous* refers to fast and efficient trajectory planning methods that can potentially execute quickly and on-board the satellite, reducing or eliminating the need for communication with the ground.

Autonomous satellite rendezvous and proximity operation trajectory design is an enabling technology for the next generation of spacecraft missions. Autonomous RPO trajectory design has been cited often by both NASA and DARPA as crucial for the future of military, civil, and commercial spacecraft; as in, for example, NASA's 2021 Technical Roadmap for the future [2]–[5].

---

[1] Graduate Student, Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech 24060
[2] Research Scientist, Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech 24060
[3] Professor, Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech 24060

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

For relative motion path planning, linear approximations of the relative motion between two satellites are typically used for speed and simplicity. The Hill-Clohessy-Wiltshire (HCW) model is perhaps the most common, these equations are a linear approximation of the circular-restricted nonlinear equations of relative motion as described in section 2.

In recent years there has been a large amount of research in optimal control for trajectory optimization, summarized in surveys by Betts, Conway, GuoQiang, and Rao, [6]–[9] and a survey by Topputo focusing on direct transcription methods for low-thrust space trajectories [10]. Additional surveys by Goerzen, L. Yang, and Y. Yang focus on random sampling and graph search methods to solve the optimal motion planning problem. This approach has been used in the robotics field to provide extremely fast solutions to high degree-of-freedom problems and has also been applied to guidance for unmanned aerial vehicles (UAVs), autonomous vehicles (2007 DARPA Urban Challenge) [11] and spacecraft trajectory planning [12]–[14]. In the sections below, the most promising solution methods will be summarized and presented along with the relevant literature.

Simple spacecraft relative motion path planning problems can be expressed as convex optimization problems and solved via simple gradient descent methods. Benefits of convex optimization include quicker, guaranteed, convergence to the global minimum without the need for an initial guess. Convex optimization is not suitable, however, for problems with complex non-linear and non-convex constraints such as those posed by complex keep out regions. Some recent efforts using Mixed Integer Linear Programming (MILP) [15], [16] and clever approximations of non-convex keep-out-ellipses as a combination of rotating hyperplanes [17], [18] allow for solving slightly more complex relative motion path planning problems. Mixed Integer Linear Programming is a solution method that splits the trajectory optimization problem into several convex regions allowing for the use of binary constraints to represent keep-out-regions.

Direct collocation methods are perhaps the most promising group of methods for solving complex satellite relative motion path planning problems. These methods discretize the continuous trajectory into state and control variables at discrete timesteps with space between the nodes being represented with polynomial splines. Constraints, including dynamical constraints, are then imposed on each of the nodes and the resulting nonlinear program (NLP) can be solved via a NLP solver. The process of converting the continuous time problem into its' discretized version is known as transcription. Many different kinds of transcription methods have been used to determine how and where nodes are chosen as well as how the state and control is approximated between nodes. Note that this approach is more costly than convex optimization, but it allows for nonlinear and non-convex constraints that better describe the satellite relative motion path planning problem. Drawbacks of using direct collocation methods are that the resulting NLP both requires and is sensitive to the quality of the initial guess used with the solver. Poor guesses may not converge to feasible solutions even if one exists and the nonlinear solver may also get stuck in local minima. For this reason, one of the main difficulties in implementing a direct collocation method is in constructing the initial guess trajectory. References [6], [7], [9], [19], [20] suggest first solving a more simple but related problem analytically or by exploiting evolutionary algorithms that are better suited to finding global minima. Reference [21] Uses a type of direct collocation known as pseudospectral or orthogonal collocation that involves the use of Legendre polynomials as global basis functions. Initial guesses for the NLP are generated by solving approximate problems involving parameter optimization of finite sequences of burns and coasts via a particle swarm optimization problem and a genetic algorithm.

Sampling and graph search path planning algorithms have been long used for motion planning in other fields and have more recently been applied to spacecraft relative motion trajectory planning. These methods utilize random (or deterministic) sampling in order to generate paths through the solution space. Properties of these algorithms allow for fast runtime and global optimality guarantees. These algorithms are extremely efficient at solving path planning problems in high dimensional spaces with complex keep out regions. A wide range of sampling-based algorithms are described in the literature with various adaptions to different kinds of problems. Anytime random sampling-based algorithms such as RRT (rapidly exploring random tree) and RRT* are included in this section. These methods continually generate samples in the state space and continuously attempt connections between nearby samples. RRT* (the * denotes asymptotic optimality) provides an improvement on RRT in that if a lower cost path becomes available through a newly sampled node, that path is added to the tree and the 'old' higher cost path is removed. In [22] an RRT* algorithm was used to generate an initial path in a complex spacecraft relative motion planning problem. This initial path was then refined using a basic local optimizer. FMT* is another sampling-based motion planning algorithm

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

which was applied to the spacecraft relative motion problem in [23]. This algorithm boasts the ability for shifting a large portion of computation off-line to better support on-board applications.

Probabilistic Road Map (PRM) takes one batch of samples and generates an interconnected graph of state variables between the start and goal nodes. Graph search algorithms such as A* and Dijkstra's algorithm are then used to search the graph for the shortest (or lowest cost) path. A* utilizes a heuristic, typically distance to the goal node, and a cost, current path length, in order to bias the search towards the goal by prioritizing the expansion of nodes that can reach the goal with lower cost. If the heuristic is well formulated (estimated cost to reach goal is never higher than actual cost to reach goal) then A* ensures optimality to the level of discretization. A* based methods have been used in the past satellite relative motion path planning problems starting with the basic rendezvous problem in [24]–[26] that similarly use a cost function composed of a fuel estimate in order to find low delta-V solutions, though obstacles are not considered. Reference [27] uses a similar A* based motion planning algorithm to solve the basic rendezvous problem with plume constraints in the presence of obstacles where new nodes are generated by randomly sampling feasible controls based on a normal distribution. Reference [28] makes some improvements on the algorithm in [27], expanding it to use the Tshauner-Hempel equations of motion and provides different weight functions for tree expansion. Reference [29] also attempts to solve the autonomous spacecraft rendezvous and docking problem by utilizing an A* search method. This paper expands on previous work to include different heuristics (minimum time and minimum path length) but makes no mention of minimum fuel trajectories. This paper improves upon collision avoidance by preemptively eliminating nodes that only lead to infeasible child nodes (collisions).

The main benefit to these methods is solution speed. While optimization methods may better approach a true optimal solution with enough time, metaheuristic randomness-based methods are often used to more quickly find an adequate (though not necessarily optimal) solution. However, many of these algorithms have asymptotic optimality in that as the number of samples taken approaches infinity the lowest cost path approaches the true optimal – though in practice this usually requires excessive computational time. Another benefit to these randomness-based sampling algorithms is the ability to avoid getting stuck in local minima (a shortcoming of potential field and gradient based methods).

1.1 Trajectory Design Constraints and Challenges

Challenges to current methods for trajectory design include the use of nonlinear objective functions (e.g., minimum fuel), collision avoidance, active & passive safety considerations, thruster limitations & thruster plume considerations, and uncertain maneuver duration. Many current solution methods rely on linear constraints and convex solution methods which are no longer applicable to more complex mission design.

The ability to handle optimization problems with these kinds of complex constraints is central to advancing rendezvous and proximity operations and enabling autonomous in-space-inspection. This motivates the design of a fast and efficient solution method capable of handling complex constraints.

1.1.1 Collision Avoidance:

Collision avoidance is the most important constraint to consider when evaluating trajectory planning algorithms for autonomous rendezvous and proximity operations. An accidental collision is the worst outcome for a spacecraft mission, causing damage or destroying one or both spacecraft involved resulting in mission failure. The ability to accurately and robustly incorporate collision avoidance constraints into a fast and efficient algorithm for trajectory planning is necessary for advancing autonomous proximity operation capabilities. This constraint is generally formulated such that:

$$r_{sc} \notin X_{obs} \tag{1}$$

Where r_{sc} denotes the position vector of the spacecraft and X_{obs} denotes a complex obstacle region within the relative frame. Obstacles in this region can be another spacecraft, orbital debris, or other area for the follower spacecraft to avoid, potentially related to sunlight constraints of the target spacecraft, sensor blind spots, etc. In this way, basic collision avoidance constraints can approximate a series of more complicated constraints.

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

It is perhaps simplest to express the keep-out-regions in X_{obs} as ellipsoids, relating to uncertainties in the position of the object or region to avoid. If needed, additional complex keep-out-regions can be constructed by combining several simple shapes though this work focuses on ellipsoids for mathematical simplicity.

For many problems, X_{obs} becomes a complex region making collision avoidance constraints non-convex and the related trajectory planning problem inherently difficult to solve.

1.1.2 Thruster Plume Impingement:

Thruster plume impingement is another complex constraint necessary for many spacecraft proximity operations. Exhaust gasses from thruster firings form a plume that can interact with nearby objects or surfaces, referred to as thruster plume impingement. Plume impingement may cause disturbing forces, unwanted heating, and contamination of target spacecraft surfaces [30], [31]. Contamination or damage to optical equipment or sensors on the target spacecraft is undesirable and necessitates a solution method capable of handling thruster plume impingement constraints. These typically take the form of restricting the thruster capability of the follower spacecraft when near the target spacecraft.

Many different models of the thruster plume exist. For this work a simple model approximating the thruster plume as a line segment with length proportional to the magnitude of the thrust is used. This thrust length, with some additional tolerance, is then checked within the collision avoidance algorithm for impact.

Simplification of the thruster plume model is required to maintain computational time limits. Incorporating an extremely accurate thruster plume model and/or exhaustively checking for collision is computationally inefficient and prohibitive for a fast and efficient algorithm capable of being autonomously run on-board.

Even a basic implementation of a thruster plume impingement constraint model is non-convex and requires a solver capable of incorporating non-convex constraints in order to generate a plume impingement free trajectory.

1.2 Research Methodology:

Two solution methods for optimal trajectory planning will be presented in this work with adaptations to the relative motion trajectory planning problem and applied to several in-space-inspection maneuvers:

1. Adapted A* Search Algorithm: path planning algorithm that utilizes a heuristic in order to efficiently search a graph of nodes for the lowest cost path from the start node to the goal node.

2. Direct Collocation: Nonlinear trajectory optimization method in which state and control variables are approximated using piecewise continuous polynomials between a set number of nodes used as inputs to a nonlinear programming solver.

2. Problem Dynamics

2.1 Relative Satellite Motion

The HCW (Hill-Clohessy-Wiltshire) equations describe the motion of a ‘follower’ or ‘deputy’ satellite relative to a ‘target’ or ‘chief’ satellite in the target satellite’s RIC reference frame, centered on the target spacecraft. These equations are a result of a linearization of the circular-restricted nonlinear equations of relative motion which are valid for relative distances (between the target and follower spacecraft) much smaller than the distance between the target and the center of the central body (Earth) and for smaller timescales.

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

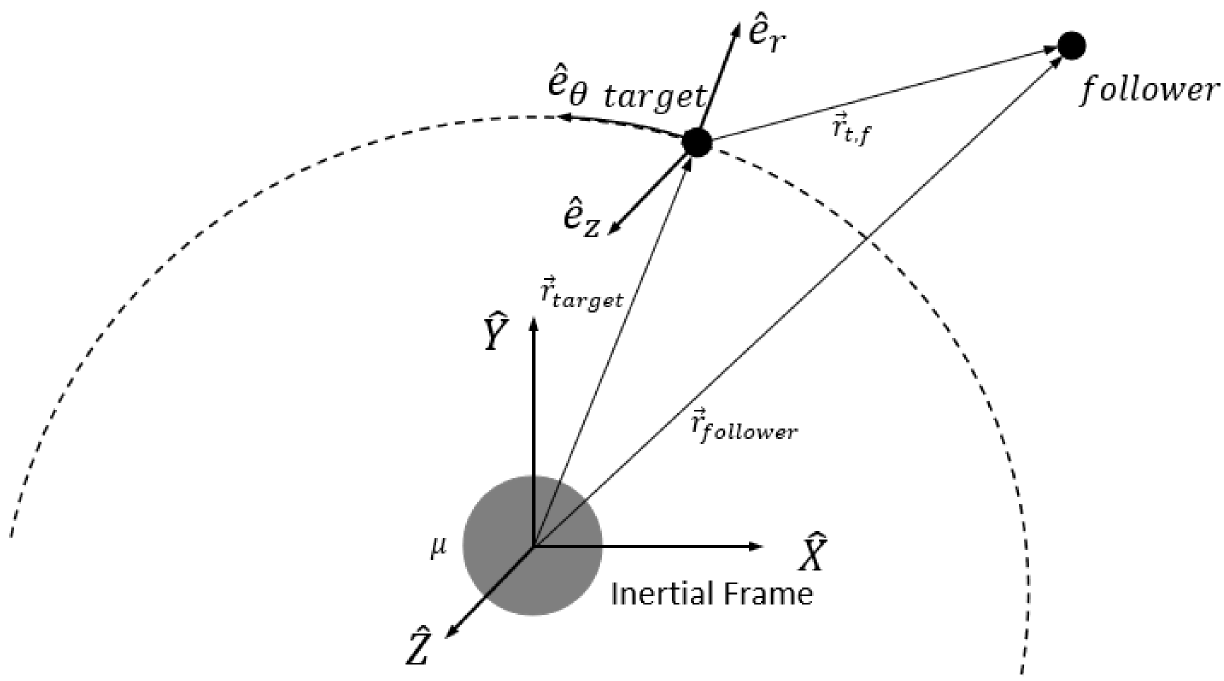

Figure 1: RIC reference frame for follower satellite in proximity operations, centered at the target.

The target satellite is restricted to be in a circular orbit and other parameters for the target's orbit are not required to represent the relative motion. The relative motion equations in the curvilinear RIC frame are as follows:

$$\begin{gathered} \delta\ddot{r} - 2a_0 n\dot{\theta}_r - 3n^2\delta r = a_x \\ a_0\ddot{\theta}_r + 2n\delta\dot{r} = a_y \\ \ddot{\phi}_r + n^2\phi_r = a_z \end{gathered} \tag{2}$$

Where $(\delta r, \theta_r, \phi_r)$ are the relative coordinates along the radial, curvilinear in-track, and curvilinear cross-track directions respectively and a_0 corresponds to the semi-major axis of the target satellite. For simplicity we set:

$$\begin{gathered} x = \delta r \\ y = a_0\theta_r \\ z = a_0\phi_r \end{gathered} \tag{3}$$

Such that the equations become:

$$\begin{gathered} \ddot{x} - 2n\dot{y} - 3n^2 x = a_x \\ \ddot{y} + 2n\dot{x} = a_y \\ \ddot{z} + n^2 z = a_z \\ where\ n = \sqrt{\frac{\mu}{a^3}} \end{gathered} \tag{4}$$

Where $\hat{x}$ points in the radial direction, $\hat{y}$ points along the orbit of the target satellite (in-track), and $\hat{z}$ points in the curvilinear out-of-plane or cross-track direction. n is the mean-motion of the target satellite's orbit and is defined as the square root of the ratio of the central body's standard gravitational parameter, $\mu = Gm$, where G is the gravitational constant, and a is the semi-major axis of the target. $a_x, a_y,$ and a_z are the components of the acceleration of the follower due to external forces (i.e., thrusting) in the curvilinear radial, in-track, and cross track directions respectively.

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Solutions to the homogenous HCW equations (for which accelerations $a_x, a_y,$ and a_z are 0, i.e., no thrust) were first derived by Clohessy and Wiltshire in [32] and can also be found in various texts such as [33] and [34]. These solutions can be expressed with the following state transition matrix (Φ) which is a function of only the transfer time, t.

$$\begin{bmatrix} \delta r(t) \\ \delta v(t) \end{bmatrix} = \Phi(\mathrm{t}) \cdot \begin{bmatrix} \delta r(t_0) \\ \delta v(t_0) \end{bmatrix} \tag{5}$$

$$\begin{pmatrix} x(t) \\ y(t) \\ z(t) \\ \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{pmatrix} = \begin{bmatrix} 4-3\cos(nt) & 0 & 0 & \frac{\sin(nt)}{n} & \frac{2\,(1-\cos(\mathrm{nt}))}{n} & 0 \\ 6(\sin(nt)-nt) & 1 & 0 & \frac{2(\cos(nt)-1)}{n} & \frac{4\sin(nt)-3nt}{n} & 0 \\ 0 & 0 & \cos(nt) & 0 & 0 & \frac{\sin(nt)}{n} \\ 3n\sin(nt) & 0 & 0 & \cos(nt) & 2\sin(nt) & 0 \\ 6n(\cos(nt)-1) & 0 & 0 & -2\sin(nt) & 4\cos(nt)-3 & 0 \\ 0 & 0 & -n\sin(nt) & 0 & 0 & \cos(nt) \end{bmatrix} \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{pmatrix} \tag{6}$$

Where t is the transfer time – the change in time between the initial state and the final state.

3. Optimal Control and Solution Methods

Trajectory planning problems for satellite RPO involve describing a desired path for the satellite in terms of state (position, velocity) and control (thrust) variables in order to meet some objective as a function of time. In a similar manner, some problems may require attitude control as well, which would expand the state variables to include some form of attitude representation (Euler angles, quaternions, modified Rodriguez parameters), their rates of change, and control torques.

In order to find the "best", or optimal, trajectory we need to select inputs to the system (our control variables) that minimize a chosen performance index, which may be total control thrust, time of flight, any other describable quantity, or some combination thereof. Additionally, we may require that our trajectory or control be subject to a series of constraints.

3.1 Direct Collocation

The trajectory optimization method applied in this work is known as direct collocation in which state and control variables are approximated using piecewise continuous polynomials between a set number of nodes. Properties of these piecewise polynomials allow for the approximation of the trajectory using algebraic equations for integration and incorporating dynamics between discretized nodes or collocation points (known as quadrature). This greatly reduces the computational time required to generate a solution but can introduce some error if higher order or nonlinear dynamics are being approximated by linear or quadratic polynomials between nodes. Additionally, constraints are only required to be satisfied at the collocation points or nodes and not in between. Accuracy can be increased by increasing the number of nodal points along the trajectory or by increasing the order of the polynomial used to approximate the trajectory between nodal points though with a tradeoff in increased computational time. In [35] Herman and Conway demonstrated that higher degree quadrature methods with fewer subintervals showed quicker computational times for the same minimum relative error. In this work, however, a simple trapezoidal method for approximating the state and control variables between collocation points is used. The motivation behind using the trapezoidal method is simplicity and to allow for the maximum number of collocation points throughout the trajectory for a given computational time. Increasing the number of collocation points is important for ensuring adherence to complex constraints such as collision avoidance and thruster plume impingement. Having more collocation points allows the algorithm to better satisfy constraints across the entire trajectory without the need for interpolating between

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

points. More information on the specifics of the constraints considered within this solution method is presented at the end of this section.

3.1.1 Transcription process:

In order to solve the optimal control trajectory optimization problem via a direct collocation method as described above the continuous-time problem is converted into a discretized version that can be solved via a nonlinear programming solver (such as MATLAB's FMINCON function [36]), this process is known as transcription. Following reference [20], first the trajectory itself is discretized into a finite set of decision variables that represent the state, position $x(t)$ and velocity $v(t)$, and control, $u(t)$, at specific points in time. This set of decision variables are known as collocation points or nodes.

$$\begin{gathered} t \rightarrow t_0, \dots, t_k, \dots, t_N \\ x(t) \rightarrow x_0, \dots, x_k, \dots, x_N \\ v(t) \rightarrow v_0, \dots, v_k, \dots, v_N \\ u(t) \rightarrow u_0, \dots, u_k, \dots, u_N \end{gathered} \tag{7}$$

Next, the continuous system dynamics are converted into a set of constraints that can be applied to the state and control variables at each of the collocation points. The trajectory (state and control) between nodes is approximated using polynomial splines which allows for computationally efficient integral calculations between nodes which is particularly important for enforcing dynamics constraints. This is where the quadrature method comes into play. Here trapezoidal quadrature (or the trapezoid rule) is used for simplicity which gives the following equations for the dynamical constraints on the state, x:

$$\begin{gathered} \dot{x} = f \\ \int_{t_k}^{t_{k+1}} \dot{x}\, dt = \int_{t_k}^{t_{k+1}} f\, dt \\ x_{k+1} - x_k \approx \tfrac{1}{2}(t_{k+1} - t_k) \cdot (f_{k+1} + f_k) \end{gathered} \tag{8}$$

and for calculating the integral term in our objective function:

$$\int_{t_0}^{t_f} \omega(x(\tau), u(\tau), \tau) d\tau \approx \sum_{k=0}^{N-1} \tfrac{1}{2}(t_{k+1} - t_k) \cdot (\omega_{k+1} + \omega_k) \tag{9}$$

Note here that the state, x_k, and control, u_k, at each grid point are decision variables in the non-linear program and $f_k = f(x_k, u_k, t_k)$ is the system dynamics at each collocation point, k.

In addition to the dynamical constraints, there may also be bounds on the state and control variables, additional linear or nonlinear constraints along the path or at the boundaries. In the direct collocation method, these constraints are generally approximated by enforcing them at the nodal points only. The bounds on the state and control variables are approximated as:

$$\begin{gathered} x_{low} \leq x_k \leq x_{upp} \\ u_{low} \leq u_k \leq u_{upp} \end{gathered} \tag{10}$$

Path constraints are approximated at each node as:

$$g(t_k, x_k, u_k) < 0 \tag{11}$$

Boundary constraints are similarly applied at the start and final nodes:

$$h\left(t_0, t_f, x_0, x_f\right) \leq 0 \tag{12}$$

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

3.2 Adapted A* Search

The application of the A* Search path planning algorithm to rendezvous and proximity operations in the relative motion frame presents two major challenges. The first and most basic lies with representing the state space of the kinodynamic problem (i.e., a motion planning problem subject to kinematic and dynamic constraints). The second major challenge is the development of a heuristic to represent the cost-to-go to the goal from any node. This section will describe the specific implementation of the adapted A* Search algorithm to the relative motion path planning problem.

3.2.1 Grid Generation Approaches

There exist two main approaches to grid generation for grid search based problems, conventional grid and dynamic grid generation.

3.2.1.1 <u>Conventional Grid Generation:</u>

Conventional grid discretization creates a uniform grid of finite cells wherein the entire state space (composed of valid positions, velocities, and times) is discretized. Node expansion is then done by evaluating neighboring nodes in the grid. This approach is impractical for the relative motion problem as it would require a very large number of nodes in order to accurately represent every possibility within the state space – many of which would represent sub-optimal or even physically unreachable states within our kinodynamic system. The extremely large number of nodes needed to represent the entire state space of the system would also drastically slow down our solver leading to computational times that are unrealistic for on-board applications. Additionally, because relative motion trajectories are naturally curved, many velocity changes (i.e., small thrusts) would be needed to pass through points in the discretization space resulting in the generation of even larger, non-optimal, delta-V values.

3.2.1.2 <u>Dynamic Grid Generation:</u>

The second approach is a more dynamic form of grid generation and node expansion based upon motion primitives and natural relative motion orbital mechanics. Nodes for expansion from the current node are chosen by applying a predefined set of impulse burns to the current node and propagating the state through a fixed timestep with the HCW state transition matrix described in the relative motion section. This method ensures that each node is dynamically feasible and represents a reachable state under bounds on impulsive thrust magnitude. This leads to far fewer nodes evaluated and much quicker computational times.

One consideration when using this method is how to determine which and how many impulsive thrusts to consider for propagating and generating new nodes. For some problems with minimal keep out regions or other constraints it may make the most sense to use only impulsive burns that are close to the optimal burn calculated with the heuristic. However, there may exist a scenario in which the problem is heavily constrained in terms of keep out regions or other constraints in which staying near the calculated optimal trajectory is infeasible and generates no paths that avoid the keep out regions even though one may exist. For heavily constrained scenarios it may be advantageous to use a wide variety of impulsive burn amplitudes and directions. In practice the number of impulsive burns to use is often decided by the computational time constraints of the algorithm and the directions and magnitudes of these burns is decided by the constraints on the specific problem.

There are two important burns to consider propagating in every case; these would be the no thrust condition corresponding to a coast phase and the burn corresponding to the minimum cost trajectory returned by the heuristic function. The no burn case is important because it implicitly allows for trajectories to be composed of burn-coast-burn sequences which are more likely to approach an optimal solution under Pontryagins Maximum Principle [37]. The optimal burn case corresponding to the minimum cost trajectory returned by the heuristic function is important for convergence – once the follower is on the optimal trajectory it will remain on this trajectory until it reaches the goal node or violates a constraint. Depending on the choice of heuristic function, determining the optimal burn may not be possible. This will be explored in the following discussion of applicable heuristic functions.

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

3.2.2 Heuristic Function

The standard straight-line Euclidian distance heuristic function is usually a bad representation of the true cost to fly a trajectory in space – which relates more to the fuel expended and is closely tied to a change in velocity (e.g., delta-V). In order to remedy this and develop a solution method to generate minimum fuel trajectories we focus on heuristic functions capable of estimating the delta-V needed to reach the goal node. In the next section we will discuss two solution methods for generating a delta-V based heuristic. The first is a 'naïve' delta-V estimate based on the CW targeting algorithm derived from the HCW state transition matrix. A second, novel, method for delta-V estimation is developed in Section 3.2.2.2 based on solving a simple convex optimization sub problem for an N-burn rendezvous.

3.2.2.1 CW Targeting Heuristic

The CW targeting heuristic is a longstanding method for determining the optimal two burn rendezvous using the HCW equations of motion and state transition matrix. This is the optimal and only two burn rendezvous for a specific transfer time. A better measure of the overall two burn optimal rendezvous can be determined by searching over possible transfer times.

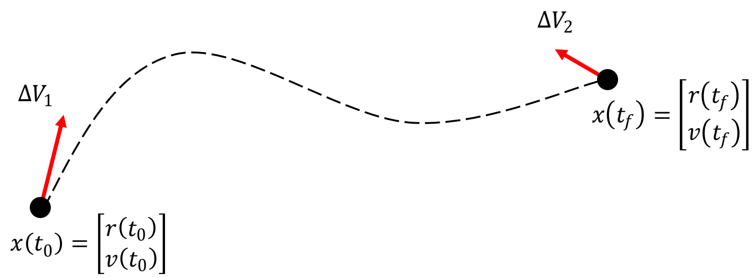

Figure 2: Visual representation of 2-burn CW Targeting algorithm. Burn magnitude and direction shown in red at each position (filled circle), resulting trajectory shown with a dashed line.

$$\begin{gathered} \Delta V_1 = {\Phi_{rv}}^{-1}\left(r(t) - \Phi_{rr}\big(r(t=0)\big)\right) - v_0^- = v_o^+ - v_0^- \\ \Delta V_2 = v_f^+ - (\Phi_{vr} r(t=0) + \Phi_{vv} v_o^+) \\ \Delta V_{total} = \Delta V_1 + \Delta V_2 \end{gathered} \tag{13}$$

This heuristic has been used in past satellite motion planning problems such as those described in the literature review section. One important note about this heuristic is that it is not admissible due to the fact that the two-burn rendezvous is not necessarily the lowest delta-V rendezvous for a fixed rendezvous time. A lower cost rendezvous can be completed with up to four different burns for the planar problem or six different burns for the full three-dimensional problem. Work by Neustadt states that for linear dynamics with an *n* dimensional state space, at most *n* impulsive burns are needed to produce an optimal transfer [38]. A more detailed explanation of various direct and indirect methods to solve for the *n* impulsive burns are given in [39]–[44]. These sources were used as a general reference for the development of the convex optimization heuristic described in the next section.

3.2.2.2 Convex Optimization Sub Problem Heuristic

The motivation behind the convex optimization sub problem heuristic is to design a delta-V heuristic that is *admissible* for use with the adapted A* Search algorithm. Admissibility of the heuristic ensures optimality to the level of discretization, known as resolution optimality, and reduces the number of nodes that need to be checked [45]. In order for the heuristic to be admissible it needs to never overestimate the actual cost to reach the goal node. One way to do this is similar to the direct method approach in [39] in which the times and delta-V values of each burn are formulated as inputs to a nonlinear optimization problem with additional nonlinear constraints to ensure that the state (relative position, propagated using the HCW linearized dynamics model) matches at each of the burn locations. This problem can be solved via a nonlinear optimizer such as MATLAB's FMINCON to give a true optimal minimum delta-V for *n* number of impulsive burns at any burn time and for any final transfer time. This nonlinear optimization solution

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

represents a true lower bound on the delta-V needed but is too complex and inefficient to be used as a heuristic within the adapted A* Search algorithm.

The adapted A* Search algorithm allows burns to occur only at a discretized and finite amount of fixed timesteps. By also constraining the final time of rendezvous we can cast the discretized optimal multi-burn rendezvous as a convex second order cone problem (SOCP) instead of a nonlinear optimization problem. This approach was shown in [46] to closely approximate the true optimal solution if enough timesteps are used. As a heuristic for the adapted A* Search algorithm, only timesteps where burns can occur need be considered in the convex optimization problem to provide an admissible and aggressive heuristic. These timesteps can be easily identified by looking at the setup of the adapted A* Search algorithm. Due to the convex nature of the problem, numerical solvers can more easily find a global optimal solution in a very short amount of time, making this heuristic potentially feasible for on-board applications. This convex optimization problem is formulated as follows:

$$\begin{gathered}
\min_{\Delta V_j, \sigma_j} J = \sum_{j=1}^{N} \sigma_j \\
\textit{Subject to}: \\
\Delta V_{x,j}^2 + \Delta V_{y,j}^2 + \Delta V_{z,j}^2 \leq \sigma_j^2 \qquad \text{for } j = 1, \ldots, N \\
x_{goal} = \Phi(\mathrm{N}\Delta\mathrm{t}) x_{start} + \sum_{j=1}^{N} \Phi((\mathrm{N}-\mathrm{j})\Delta\mathrm{t}) \begin{pmatrix} \mathbf{0} \\ \Delta V_j \end{pmatrix} \\
\sigma_{low} \leq \sigma_j \leq \sigma_{upp}
\end{gathered} \tag{14}$$

Here we include an additional set of slack variables, σ_j, corresponding to the magnitudes of each of the N burns. These variables allow us to express the originally nonlinear objective function as linear, a necessary condition for formulation as a SOCP. In this way we create a "relaxed" convex optimization problem whose solution is the same as if we had used the nonlinear 2-norm minimum delta-V objective function [17], [18], [47]. This approach allows us to minimize the total delta-V expenditure where each burn is represented by a single thruster. The only other variables in the solver are then the impulsive burns themselves where:

$$V_j = \begin{bmatrix} \Delta V_{x,j} \\ \Delta V_{y,j} \\ \Delta V_{z,j} \end{bmatrix} \tag{15}$$

The timesteps between burns, $\Delta\mathrm{t}$, start state, x_{start}, goal state, x_{goal}, are known a priori. We can then calculate a minimum delta-V for an N-burn impulsive rendezvous between a start state and goal state. This delta-V value is the same as our objective function:

$$\Delta V_{total} = \sum_{j=1}^{N} \sigma_j \tag{16}$$

This value can then be used as an admissible heuristic within our A* path planning algorithm, improving convergence, and guaranteeing optimality to the level of discretization. The algorithm is additionally said to be complete in that if the algorithm finishes in a fixed time and returns a solution if one exists [14]. These are very important properties for our autonomous relative motion planning problem and justify the use of a more computationally expensive heuristic estimate. Additional information and proofs for resolution optimality and completeness of the A* algorithm can be found in [45].

A visualization of the convex optimization sub problem can be seen in *Figure 3*.

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

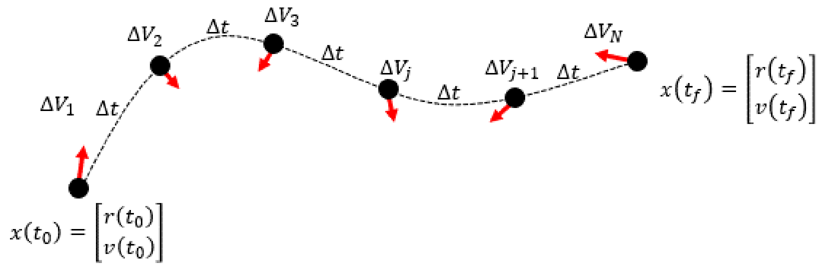

Figure 3: Visual representation of Convex Optimization Multi-Burn Rendezvous algorithm. Burn magnitude and direction shown in red at each position (filled circle), resulting trajectory shown with a dashed line.

4. Simulation Results

Simulations for several test cases were performed in order to compare the two algorithms in terms of computational time and solution quality (in terms of delta-V). All simulations were run in MATLAB 2021a on an Intel(R) Core(TM) i7-6600U CPU processor at 2.60 GHz and with 16 GB of RAM.

4.1 Waypoint Maneuver: Simple Keep Out Region

The first example problem deals with a simple waypoint maneuver proceeding from 'behind' the target in the in-track direction to 'in-front' of the target in the in-track direction with an ellipsoid keep out region centered at (0,0,0). Initially the follower is at (0,-20,0) and is tasked with reaching the goal node at (0,20,0) within a fixed amount of time while minimizing fuel. Additional problem properties are listed in Table 1. Max thrusts for the impulsive adapted A* Search and Direct Collocation solution methods were developed such that the maximum thrust impulsive thrust corresponds to the sum of the maximum continuous thrusting over the same interval to provide some parity between solutions.

Table 1: Problem Properties for Simple Waypoint Maneuver

Parameter	Chief Orbit (km)	Max Transfer Time (s)	Follower Mass (kg)	Max Thrust (impulsive) (m/s)	Max Thrust (continuous) (m/s^2)
Value	6791	600	1	.06	.001

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Generated trajectories usig the two solution methods described above can be seen in Figure 4. The adapted A* Search algorithm solution is plotted on the left while the direct collocation nonlinear optimization method solution is shown on the right.

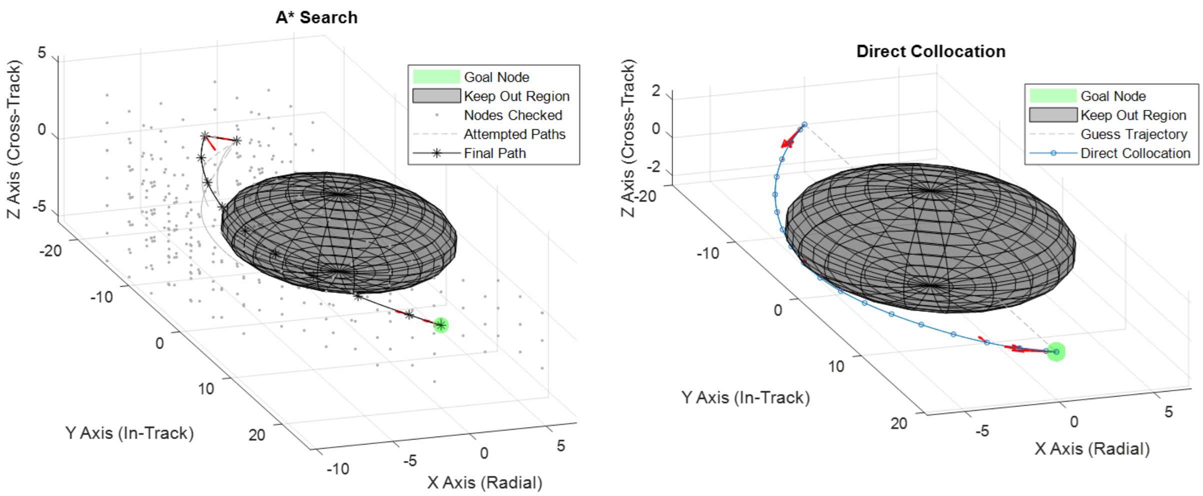

Figure 4: Trajectory solutions to the simple waypoint maneuver example problem. (Left) Trajectory generated via the Adapted A Search algorithm. (Right) Trajectory generated via the Direct Collocation nonlinear optimization method.*

As can be seen above, the two trajectories are quite similar. The computational solution time and total delta-V values for each solution method are listed in Table 2. As expected, the direct collocation nonlinear optimization solution results in a lower delta-V value but requires substantially more computational time. The adapted A* Search algorithm generates a solution that is quite close to optimal in a fraction of the solution time.

Table 2: Solution Results for Simple Waypoint Maneuver

Solution Method	**Solution time (s)**	**Total Delta-V (m/s)**
Adapted A* Search	3.54	0.2034
Direct Collocation	8.17	0.1752

4.2 Waypoint Maneuver: Complex Keep Out Region with Thruster Plume Constraints

The second example problem deals with a more complex waypoint maneuver from an initial location of (-20,-20,-5) to a final location of (0,0,0) with three ellipsoid keep out regions and thruster plume impingement constraints. For this example, the thruster plume was modelled as a line segment extending from the location of the burn in the direction of the plume, scaled by the magnitude of the plume to a maximum of 10 meters (corresponding to maximum thrust at a node). For the Direct Collocation solution method the plume impingement is checked only at nodal points though continuous thrusting is employed. This approximation is necessary to keep computational time low. Additional problem properties are listed in Table 3.

Table 3: Problem Properties for Complex Waypoint Maneuver

Parameter	Chief Orbit (km)	Max Transfer Time (s)	Follower Mass (kg)	Max Thrust (impulsive) (m/s)	Max Thrust (continuous) (m/s^2)
Value	6791	2400	1	.03	.0002375

Trajectories generated using the two solution methods described above can be seen in Figure 5. The adapted A* Search algorithm solution is plotted on the left while the direct collocation nonlinear optimization method solution is shown on the right.

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

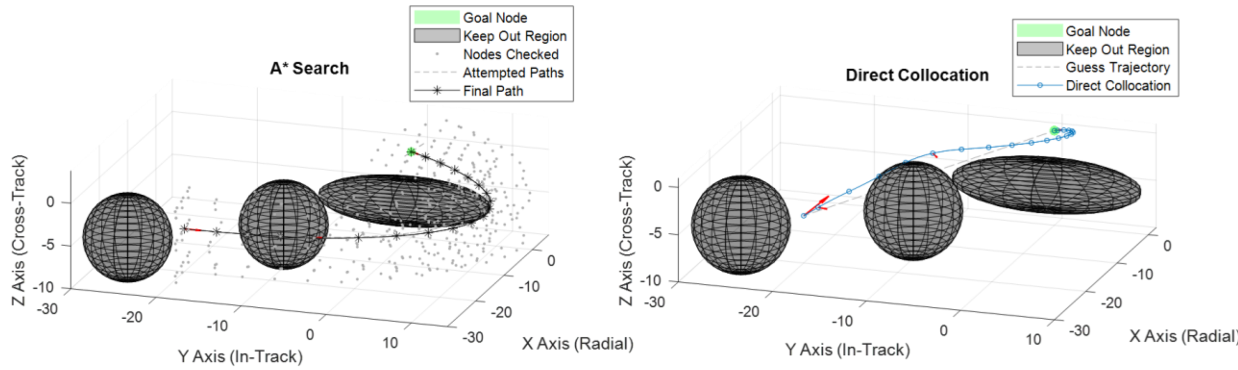

Figure 5: Trajectory solutions to the waypoint maneuver with complex keep out regions and thruster plume impingement constraints. (Left) Trajectory generated via the Adapted A Search algorithm. (Right) Trajectory generated via the Direct Collocation nonlinear optimization method.*

Again, the two solution methods result in similar trajectories. The adapted A* Search algorithm performs quite well with complex constraints and provides a comparable solution in terms of delta-V while maintaining a short computational solution time.

Table 4: Solution Results for Complex Waypoint Maneuver

Solution Method	**Solution time (s)**	**Total Delta-V (m/s)**
Adapted A* Search	6.91	0.0708
Direct Collocation	32.04	0.0627

4.3 Injection into NMT Ellipse

This example problem deals with an injection into a natural motion trajectory. The follower attempts to enter into a bounded trajectory about the target, useful for long term station keeping and inspection. This bounded trajectory takes the form of a 2-1 ellipse in the in-track and radial directions respectively. Additional problem properties are listed in Table 5.

Table 5: Problem Properties for Injection into NMT

Parameter	Chief Orbit (km)	Max Transfer Time (s)	Follower Mass (kg)	Max Thrust (impulsive) (m/s)	Max Thrust (continuous) (m/s^2)
Value	6791	2400	1	.03	.0002375

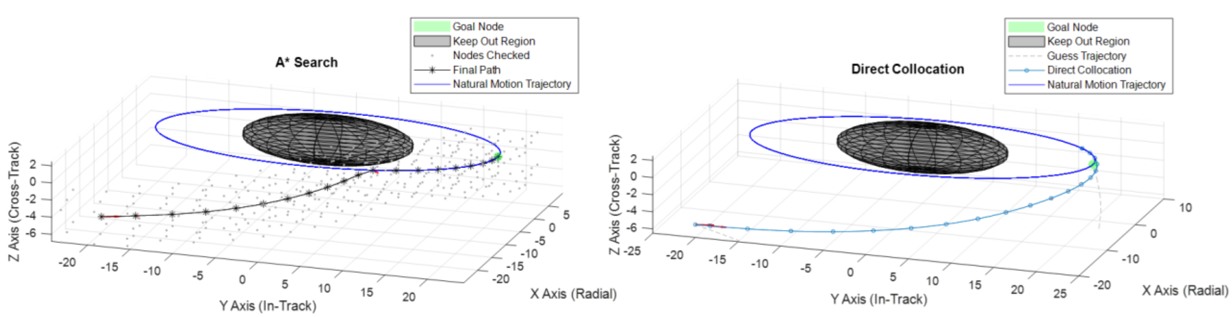

Figure 6: Trajectory solutions to natural motion trajectory example. (Left) Trajectory generated via the Adapted A Search algorithm. (Right) Trajectory generated via the Direct Collocation nonlinear optimization method.*

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Again, the two trajectories are quite similar. For this case, the adapted A* Search method outperforms the nonlinear optimization-based Direct Collocation method in both total delta-V and solution time. This could be due to the nonlinear optimizer getting stuck in a local minimum near its original guess trajectory or from the slight disparity between impulsive and continuous thrusting maximum limits. Exact values for this solution case can be found in Table 6.

Table 6: Solution Results for Natural Motion Trajectory Injection

Solution Method	Solution time (s)	Total Delta-V (m/s)
Adapted A* Search	5.57	0.0486
Direct Collocation	24.57	0.0499

5. Conclusion

Both the Adapted A* Search algorithm and the nonlinear optimization based direct collocation method for trajectory generation are applicable to the satellite relative motion problem for in-space-inspection. Both solution methods are capable of incorporating complex constraints such as multiple complex keep out regions and thruster plume impingement. Generally, the adapted A* Search algorithm had significantly lower computation times and the nonlinear optimization direct collocation method had slightly lower delta-V values (though not in the example above).

The Adapted A* Search method is quite fast depending on the number of nodal points used and the number of delta-V burns to propagate for child nodes. The Adapted A* Search algorithm also has some nice convergence properties including *resolution completeness* (if a solution exists in our discretization, it will be found in finite time) and global optimality, again within our level of discretization.

The nonlinear optimization based direct collocation method is capable of incorporating almost any constraint (including but not limited to the keep out regions and thruster plume impingement) and capable of numerically finding a solution via a nonlinear programming solver. The nonlinear programming solver, however, is not able to provide any guarantees on finding a solution (i.e., returning a solution if one exists) or ensuring that the solution found is optimal.

For autonomous application on-board, either method or a combination of the two can be applied depending on the specific goals of the mission. An initial solution generated by the adapted A* Search algorithm can be used to seed the Direct Collocation method which, if time permits, may offer a better overall solution. Both methods were shown to generate good trajectories with fairly low computational times that may allow for on-board application.

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

6. REFERENCES

[1] W. Fehse, *Automated Rendezvous and Docking of Spacecraft*, vol. 1, no. 1. 2003.

[2] M. E. Polites, "An Assessment of the Technology of Automated Rendezvous and Capture in Space," 1998.

[3] M. E. Polites, "Technology of automated rendezvous and capture in space," *J. Spacecr. Rockets*, vol. 36, no. 2, pp. 280–291, 1999.

[4] D. A. Whelan, E. A. Adler, S. B. Wilson III, and G. M. Roesler, Jr., "DARPA Orbital Express program: effecting a revolution in space-based systems," in *Small Payloads in Space*, 2000, vol. 4136, no. November 2000, pp. 48–56.

[5] "2020 NASA Technology Taxonomy," 2020.

[6] J. T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *J. Guid. Control. Dyn.*, vol. 21, no. 2, pp. 193–207, 1998.

[7] B. A. Conway, "A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems," *J. Optim. Theory Appl.*, vol. 152, no. 2, pp. 271–306, 2012.

[8] G. Q. Huang, Y. P. Lu, and Y. Nan, "A Survey of Numerical Algorithms for Trajectory Optimization of Flight Vehicles," *Sci. China Technol. Sci.*, vol. 55, no. 9, pp. 2538–2560, 2012.

[9] A. V. Rao, "A Survey of Numerical Methods for Optimal Control," *Adv. Astronaut. Sci.*, vol. 135, 2010.

[10] F. Topputo and C. Zhang, "Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications," *Abstr. Appl. Anal.*, vol. 2014, 2014.

[11] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path Planning for Autonomous Driving in Unknown Environments," in *Experimental Robotics*, 2009, pp. 55–64.

[12] C. Goerzen, Z. Kong, and B. Mettler, "A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance," *J. Intell. Robot. Syst. Theory Appl.*, vol. 57, no. 1–4, pp. 65–100, 2010.

[13] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, "Survey of Robot 3D Path Planning Algorithms," *J. Control Sci. Eng.*, vol. 2016, 2016.

[14] Y. Yang, J. Pan, and W. Wan, "Survey of Optimal Motion Planning," *IET Cyber-Systems Robot.*, vol. 1, no. 1, pp. 13–19, 2019.

[15] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft Trajectory Planning with Avoidance Constraints using Mixed-Integer Linear Programming," *J. Guid. Control. Dyn.*, vol. 25, no. 4, pp. 755–764, 2002.

[16] A. Richards and J. P. How, "Model Predictive Control of Vehicle Maneuvers with Guaranteed Completion Time and Robust Feasibility," *Proc. Am. Control Conf.*, vol. 5, pp. 4034–4040, 2003.

[17] N. G. Ortolano, "Autonomous Trajectory Planning for Satellite RPO and Safety of Flight Using Convex Optimization," Utah State University, 2018.

[18] N. Ortolano, D. K. Geller, and A. Avery, "Autonomous Optimal Trajectory Planning for Orbital Rendezvous, Satellite Inspection, and Final Approach Based on Convex Optimization," *J. Astronaut. Sci.*, pp. 444–479, 2021.

[19] B. A. Conway, "The Problem of Spacecraft Trajectory Optimization," in *Spacecraft Trajectory Optimization*, B. A. E. Conway, Ed. Cambridge University Press, 2010, pp. 1–15.

[20] M. Kelly, "An Introduction To Trajectory Optimization: How To Do Your Own Direct Collocation," *SIAM Rev.*, vol. 59, no. 4, pp. 849–904, 2017.

[21] E. R. Prince, "Optimal Finite Thrust Guidance Methods for Constrained Satellite Proximity Operations Inspection Maneuvers," Air Force Institute of Technology, 2018.

[22] L. J. Digirolamo, "A Hybrid Stochastic Motion Planning Algorithm for Safe and Efficient, Close Proximity, Autonomous Spacecraft Missions," Pennsylvania State University, 2014.

[23] J. A. Starek, "Sampling-Based Motion Planning for Safe and Efficient Spacecraft Proximity Operations," Stanford University, 2016.

[24] C. K. Niiya, "An Application of the A* Search to Trajectory Optimization," Massachusetts Institute of Technology, 1990.

[25] J. F. Raquet, "Six Degree of Freedom Trajectory Planner for Spacecraft Proximity Operations Using an A* Node Search," Massachusetts Institute of Technology, 1991.

[26] M. C. Jackson, "A Six Degree of Freedom Plume-Fuel Optimal Trajectory Planner for Spacecraft Proximity Operations Using an A* Node Search," Massachusetts Institute of Technology, 1994.

[27] J. M. Phillips, L. E. Kavraki, and N. Bedrossian, "Probabilistic Optimization Applied to Spacecraft Rendezvous and Docking," in *Advances in the Astronautical Sciences*, 2003, vol. 114 I, pp. 261–274.

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

[28] G. Saive and M. Vasile, “Probabilistic Optimisation applied to Spacecraft Rendezvous on Keplerian Orbits,” 2004.
[29] G. D. Francis, E. G. Collins, O. Chuy, and A. Sharma, “Sampling-Based Trajectory Generation for Autonomous Spacecraft Rendezvous and Docking,” in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013.
[30] W. Fehse, “Close Range Rendezvous Operations,” in *Automated Rendezvous and Docking of Spacecraft*, 1st ed., M. J. Rycroft and W. Shyy, Eds. New York: Cambridge University Press, 2003, pp. 19–24.
[31] G. Dettleff, “Plume Flow and Plume Impingement in Space Technology,” *Prog. Aerosp. Sci.*, vol. 28, no. 1, pp. 1–71, 1991.
[32] W. H. Clohessy and R. S. Wiltshire, “Terminal Guidance System for Satellite Rendezvous,” *J. Aerosp. Sci.*, vol. 27, no. 9, pp. 653–658, 1960.
[33] D. A. Vallado and W. D. McClain, *Fundamentals of Astrodynamics and Applications*. Springer Netherlands, 2001.
[34] W. E. Wiesel, *Spaceflight Dynamics*, 3rd ed. CreateSpace, 2010.
[35] A. L. Herman and B. A. Conway, “Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules,” *J. Guid. Control. Dyn.*, vol. 19, no. 3, pp. 592–599, 1996.
[36] “Fmincon,” *MathWorks*, 2021. [Online]. Available: https://www.mathworks.com/help/optim/ug/fmincon.html. [Accessed: 06-Apr-2021].
[37] A. C. Rogers *et al.*, “Optimization-Based Guidance For Satellite Relative Motion,” 2016.
[38] L. W. Neustadt, “Optimization, A Moment Problem, and Nonlinear Programming,” *SIAM Control*, vol. 2, no. 1, pp. 33–53, 1964.
[39] S. P. Hughes, L. M. Mailhe, and J. J. Guzman, “A Comparison of Trajectory Optimization Methods for the Impulsive Minimum Fuel Rendezvous Problem,” in *26th Annual Guidance and Control Conference*, 2003.
[40] J. E. Prussing, “Optimal Four-Impulse Fixed-Time Rendezvous in the Vicinity of a Circular Orbit,” *AIAA J.*, vol. 7, no. 5, pp. 928–935, 1969.
[41] J. E. Prussing, “Optimal Two- and Three-Impulse Fixed-Time Rendezvous in the Vicinity of a Circular Orbit,” *AIAA J.*, vol. 8, no. 7, pp. 1221–1228, 1970.
[42] J. E. Prussing and J. H. Chiu, “Optimal Multiple-Impulse Time-Fixed Rendezvous Between Circular Orbits,” *J. Guid. Control. Dyn.*, vol. 9, no. 1, pp. 17–22, 1986.
[43] T. E. Carter, “Fuel-Optimal Maneuvers of a Spacecraft Relative to a Point in Circular Orbit,” *J. Guid. Control. Dyn.*, vol. 7, no. 6, pp. 710–716, 1984.
[44] T. E. Carter, “Quadratic-Based Computation of Four-Impulse Optimal Rendezvous near Circular Orbit,” vol. 23, no. 1, pp. 109–117, 2000.
[45] P. E. Hart, N. J. Nilsson, and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *IEEE Trans. Syst. Sci. Cybern.*, no. 2, pp. 100–107, 1968.
[46] B. Benedikter and A. Zavoli, “Convex Optimization of Linear Impulsive Rendezvous,” *arXiv Optim. Control*, pp. 1–13, 2019.
[47] P. Lu and X. Liu, “Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization,” *J. Guid. Control. Dyn.*, vol. 36, no. 2, pp. 375–389, 2013.

Copyright © 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com