Journal of Guidance, Control, and Dynamics
Vol. 38, No. 4, April 2015

Safe Positively Invariant Sets for Spacecraft Obstacle Avoidance

Avishai Weiss* and Christopher Petersen[†]
University of Michigan, Ann Arbor, Michigan 48109-2140
Morgan Baldwin[‡] and R. Scott Erwin[§]
U.S. Air Force Research Laboratory, Kirtland Air Force Base, New Mexico 87117-5776
and
Ilya Kolmanovsky[¶]
University of Michigan, Ann Arbor, Michigan 48109-2140

DOI: 10.2514/1.G000115

This paper presents an obstacle avoidance method for spacecraft relative motion control. In this approach, a connectivity graph is constructed for a set of relative frame points, which form a virtual net centered around a nominal orbital position. The connectivity between points in the virtual net is determined based on the use of safe positively invariant sets for guaranteed collision free maneuvering. A graph search algorithm is then applied to find a maneuver that avoids specified obstacles and adheres to specified thrust limits. As compared to conventional open-loop trajectory optimization, this approach enables the handling of bounded disturbances, which can represent the effects of perturbing forces and model uncertainty, while rigorously guaranteeing that nonconvex and possibly time-varying obstacle avoidance constraints are satisfied. Details for handling a single stationary obstacle, multiple stationary obstacles, moving obstacles, and bounded disturbances are reported and illustrated with simulation case studies.

Nomenclature

A	=	discrete-time system matrix
A_c	=	continuous-time system matrix
$\bar{A}$	=	closed-loop system matrix
B	=	discrete-time input matrix
B_c	=	continuous-time input matrix
C	=	positively invariant ellipsoidal set scaled by ρ
C_k	=	contractive ellipsoidal set for moving obstacles
$\bar{C}$	=	positively invariant ellipsoidal set defined by r and K
$\mathcal{D}$	=	union of obstacles
e_i	=	vertices of constant polytope
F_x, F_y, F_z	=	external force coordinates
$\boldsymbol{F}$	=	external force vector
int	=	interior
K	=	controller gain
$\mathcal{K}$	=	finite set of controller gains
$\mathcal{L}$	=	Lagrangian
m_c	=	spacecraft mass
$\mathcal{N}$	=	finite set of prescribed spacecraft relative positions
n	=	mean motion
O	=	set of obstacle locations q defined by Q
Q	=	shape matrix of obstacle
q	=	relative position of obstacle
$\boldsymbol{R}$	=	spacecraft position vector with respect to center of Earth
$\boldsymbol{R}_0$	=	nominal orbital position vector
r	=	relative position vector
S	=	state-to-position matrix
U	=	control vector
$u_{\max}$	=	maximum thrust
V	=	Lyapunov function
W	=	disturbance bounding set
w	=	disturbance
X	=	state vector
$X_e(r)$	=	equilibrium state set
x, y, z	=	spacecraft relative coordinates
$\gamma_{\min}$	=	minimum scaling parameter for positive invariance under disturbance
ΔT	=	sampling period
$\delta\boldsymbol{r}$	=	spacecraft position vector with respect to target location
λ_1, λ_2	=	Lagrange multipliers
μ	=	gravitational parameter
ρ	=	scaling parameter
ρ_g	=	obstacle-dependent growth distance
ρ_u	=	maximum thrust growth distance
ρ^*	=	thrust-limited growth distance
$\bigoplus$	=	Minkowski sum

I. Introduction

THE need for spacecraft obstacle avoidance techniques has grown in recent years. Obstacles of concern in space applications include, for instance, orbital debris. In fact, as many as 40% of ground trackable objects are pieces of debris that originated from explosions that now number approximately five per year [1]. Other obstacles may include appendages and spacecraft components that must be avoided during proximity operations and docking. Therefore, spacecraft maneuvering algorithms must address obstacle avoidance requirements, which result in nonconvex and, in the case of moving obstacles, time-varying constraints on spacecraft motion.

Although obstacle avoidance is a standard problem in robotics ([2] p. 66, [3]), the related spacecraft problem has several unique features. First, the space environment is relatively uncluttered, permitting a variety of maneuvers that may be optimized with respect to fuel consumption, maneuver time, etc. Second, spacecraft dynamics are quite different from those of typical robots and are affected by unmodeled perturbing forces and other sources of model uncertainty. Third, the states of the spacecraft and the obstacles can only be estimated, often with significant estimation error. Finally, given limited onboard computing power and the presence of moving objects, computational algorithms must be fast. These unique features provide motivation for the development of specialized algorithms.

Received 25 June 2013; revision received 23 March 2014; accepted for publication 27 March 2014; published online 4 September 2014.

*Graduate Student, Department of Aerospace Engineering.

[†] Graduate Student, Department of Aerospace Engineering. Member AIAA.

[‡] Research Aerospace Engineer. Member AIAA.

[§] Principal Research Aerospace Engineer. Associate Fellow AIAA.

[¶] Professor, Department of Aerospace Engineering. Member AIAA.

Interest in spacecraft trajectory optimization with obstacle avoidance has increased in recent years. An optimal control problem with path constraints constructed as keep out zones to avoid obstacles was formulated in [4]. The Sparse Optimal Control Software (SOCS) was then used to solve the problem ([5] pp. 177–180). Another nonlinear optimal control formulation was used in [6] to solve for minimum-fuel rendezvous between a target and chaser, where collision avoidance requirements were incorporated as inequality constraints. This method involved solving a sequence of unconstrained optimal control problems, whose solution converges to the solution of the original problem. A three-dimensional static cost optimization over final relative position and time-of-flight for obstacle avoidance is presented in [7], where feedback is incorporated by replanning over either constant or variable time intervals.

Obstacle avoidance strategies have also been defined using collision avoidance probabilities. Collision avoidance strategies based upon the number of evasive maneuvers, expected risk reduction, false alarm rate, required propellant consumption, and mass fraction for an accepted collision probability are presented in [8].

Guidance based on an artificial potential function is used in [7,9] to determine a rendezvous path free of obstacles. A potential function is developed with the intent that a minimum occurs at a desired relative position and then a dynamic control law is used to ensure the trajectory is collision free [9].

The spacecraft obstacle avoidance problem has also been treated using linear programming techniques [10–13]. In [10], a minimum-fuel avoidance problem is formulated with linear constraints and discrete-time dynamics. In [11], the problem is formulated as a linear program capable of including operational constraints, which results in the optimal number of maneuvers. In [12], a mixed-integer linear program resulting from combining collision avoidance, trajectory optimization, and fleet assignment is solved to obtain the optimal solution for spacecraft maneuvers. A robust linear programming technique is proposed in [13], from which a maneuver can be constructed by solving a linear programming problem with no integer constraints, guaranteeing collision avoidance with respect to bounded navigation uncertainty.

In the present paper, the development of an onboard maneuver planning approach is based on the use of safe positively invariant sets [14–17]. These sets determine connectivity between static points, that is, with zero velocity, in Hills' relative motion frame ([18] p. 299). A collection of points forms a virtual net in the vicinity of the spacecraft, where two points are said to be connected if a linear-quadratic (LQ) feedback gain can be chosen such that the transition between them satisfies thrust limits and avoids collision with potentially moving obstacles. A connectivity graph, or node adjacency matrix, for all the points in the net is constructed based on fast growth distance computations between ellipsoidal sets, whereas real-time graph search algorithms are used to optimize a point command sequence that avoids obstacle collisions. Note that points are used as artifices in maneuver construction; the spacecraft does not reach or remain at any of the points (except for the target point), switching immediately to the next point in the sequence once suitable switching conditions are satisfied.

The unique features of this problem — that is, the dynamics of spacecraft relative motion being different from that of ground robots, the nonconvex and time-dependent constraints imposed due to moving obstacles, and the handling of unmeasured disturbances and uncertainties that may in part account for navigation errors — significantly impede the application of existing open-loop trajectory optimizers, previous optimal control results, and feedback control algorithms. The present paper provides a feasible solution to this complex problem by reducing it to an online graph search, which is not conceptually or computationally complex. It relies on positively invariant set techniques similar to well-established ones reported in ([14], [16] pp. 110, 271–335, [17,19,20]) and assures robustness to unmeasured (but set-bounded) disturbances and uncertainties through mechanisms similar to those employed in robust model predictive control (MPC) [21–23]. Computationally, the approach relies on standard numerical algorithms for graph search, which are used in many applications, and whose complexity can be managed by adjusting the number of nodes in the virtual net. For other uses of graph-theoretic ideas in spacecraft maneuvering, see [24–27].

Details of online and offline computations are carefully addressed to ensure that the presented approach has low computational overhead and strong potential to be real-time implementable onboard a spacecraft. Maneuver efficiency in terms of time or fuel consumption can be improved within the framework by selecting appropriate costs for the node adjacency matrix and by employing a multigain controller switching architecture, where selection of the controller gain is part of the optimization. It is also expected that the feasible solution provided by this method can be used as an initial guess for open-loop trajectory optimizers or robust MPC algorithms. In problems of this kind, a good initial guess is critical to achieve numerical convergence within the available time. Finally, note that this solution, unlike an open-loop trajectory, does not require precise assignment of spacecraft control inputs vs time, instead relying on feedback and switching conditions for executing with increased robustness.

Preliminary versions of some results in this paper have appeared in various conference publications [28,29]. The differences with previous work [14,15,20] stem from extending that approach to handle nonconvex and time-varying obstacle avoidance constraints. Ong and Gilbert [30] motivated the use of growth distances; however, in [30], these distances are defined between polyhedral sets, whereas here ellipsoidal sets are considered.

The paper is organized as follows. In Sec. II, the relative motion model is summarized. In Sec. III, the main features of the approach, i.e., the virtual net, a multigain LQ controller, and positively invariant sets, are introduced. To introduce the framework without overloading the reader with ancillary details, the case of a single stationary obstacle is treated first in Sec. IV. Procedures to determine point connectivity and construct the point connectivity graph are developed. It is shown that the maneuver construction procedures reduce to a conventional graph search. The approach is expanded in Sec. V to handle moving obstacles either by covering the obstacle's path with a union of stationary obstacles (in the case of a fast moving obstacle) or by accounting for the motion of the obstacle in time (in the case of a slow moving obstacle). In the latter case, sets satisfying a contractivity property are employed, and the connectivity graph vertices are extended with time information. Finally, the treatment of bounded disturbances is the subject of Sec. VI. Section VII presents simulations that illustrate these approaches for avoiding both stationary and moving obstacles, and for the case of bounded disturbances. Finally, Sec. VIII contains concluding remarks.

II. Relative Motion Model

The spacecraft relative motion model presents spacecraft dynamics in the (noninertial) Hill's frame, where the origin is a target location on a nominal circular orbit.

A. Nonlinear Equations of Motion

The relative position vector of a spacecraft with respect to a target location on an orbit is expressed as

$$\delta \boldsymbol{r} = x\hat{i} + y\hat{j} + z\hat{k}$$

where x, y, and z are the components of the spacecraft's position vector relative to the target location and $\hat{i}$, $\hat{j}$, and $\hat{k}$ are the unit vectors of Hill's frame. Hill's frame has its x axis oriented along the orbital radius, y axis orthogonal to the x axis and in the orbital plane, and z axis orthogonal to orbital plane.

The position vector of the spacecraft with respect to the center of the Earth is given by $\boldsymbol{R} = \boldsymbol{R}_0 + \delta\boldsymbol{r}$, where $\boldsymbol{R}_0$ is the nominal orbital position vector. The nonlinear equation of motion for the spacecraft (relative to an inertial frame) is given by

$$\ddot{\boldsymbol{R}} = -\mu\frac{\boldsymbol{R}}{R^3} + \frac{1}{m_c}\boldsymbol{F} \tag{1}$$

where $\boldsymbol{F}$ is the vector of external forces applied to the spacecraft, $R = |\boldsymbol{R}|$, m_c is the mass of the spacecraft, and μ is the gravitational parameter.

B. Linearized Clohessy–Wiltshire–Hill Equations

For $\delta r \ll R$, the linearized Clohessy–Wiltshire–Hill (CWH) equations [31] approximate the relative motion of the spacecraft on a circular orbit as

$$\ddot{x} - 3n^2 x - 2n\dot{y} = \frac{F_x}{m_c}, \quad \ddot{y} + 2n\dot{x} = \frac{F_y}{m_c}, \quad \ddot{z} + n^2 z = \frac{F_z}{m_c} \tag{2}$$

where F_x, F_y, F_z are components of the external force vector (excluding gravity) acting on the spacecraft, $n = \sqrt{\mu/R_0^3}$ denotes the mean motion of the nominal orbit, and R_0 is the nominal orbital radius. The linearized dynamics account for differences in gravity between the spacecraft and nominal orbital location, and for relative motion effects.

The spacecraft relative motion dynamics in the orbital plane (x and y) and in the out-of-orbital plane (z) are decoupled. The in-plane dynamics are Lyapunov unstable (two eigenvalues at the origin and two eigenvalues on the imaginary axis at $\pm nj$), whereas the out-of-plane dynamics are Lyapunov stable (two eigenvalues on the imaginary axis at $\pm nj$). The in-plane dynamics are completely controllable from F_y input but are not controllable from F_x input. The out-of-plane dynamics are controllable from F_z input. These dynamics are clearly very different from typical ground robots.

Assuming a sampling period of ΔTs, the model (2) can be converted to a discrete-time form

$$X(t+1) = AX(t) + BU(t) \tag{3}$$

where $X(t) = [x(t), y(t), z(t), \dot{x}(t), \dot{y}(t), \dot{z}(t)]^T$ is the state vector at the time instant $t \in Z^+$, $U(t) = [F_x(t), F_y(t), F_z(t)]^T$ is the control vector of thrust forces at the time instant $t \in Z^+$, and $A = \exp(A_c \Delta T)$, $B = \int_0^{\Delta T} \exp(A_c(\Delta T - \tau))\,\mathrm{d}\tau B_c$ are the discretized matrices obtained based on the continuous-time system realization (A_c, B_c) in Eq. (2).

Note that alternative formulations of the control vector are possible. For instance, it may be desirable to prescribe Δv commands (realized by thrust forces) ([32] pp. 271–279) or employ on–off thruster time width modulation. These alternative formulations also lead to a model in the form of Eq. (3); however, the B matrix is numerically different. For instance, the Δv formulation has the same A matrix as given previously, but the B matrix is instead given by

$$B = e^{A_c \Delta T} \begin{bmatrix} 0_{3\times 3} \\ I_{3\times 3} \end{bmatrix} \tag{4}$$

III. Preliminaries

In this section, the basic features of the framework, specifically, the virtual net, the multigain LQ controller, and positively invariant sets, are reviewed.

A. Virtual Net

The virtual net comprises a finite set of points $X_e(r)$ corresponding to a finite set of prescribed spacecraft relative positions $r \in \mathcal{N} = \{r_1, r_2, \ldots, r_n\} \subset R^3$

$$X_e(r_k) = [r_k \;\; 0]^T = [r_{x,k} \;\; r_{y,k} \;\; r_{z,k} \;\; 0 \;\; 0 \;\; 0]^T, \quad k = 1, \cdots, n \tag{5}$$

whose velocity states are zero and where n is the number of points in the virtual net (see Fig. 1). The dots in Fig. 1 correspond to positions r_k, $k = 1, \ldots, n$, at points $X_e(r)$ on a virtual net, whereas the ellipsoid represents the obstacle position and uncertainty. It is assumed that for all $r \in \mathcal{N}$, the corresponding values of control necessary to support the specified point in steady state satisfy the imposed thrust limits.

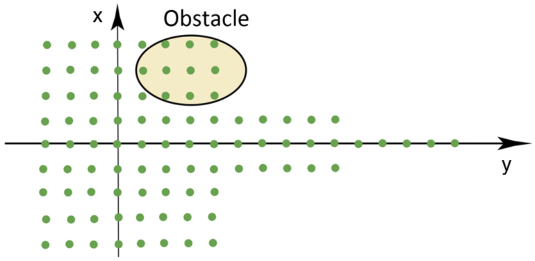


Fig. 1 Virtual net for obstacle avoidance.

Although the virtual net is based on positions corresponding to constant control inputs, it may be possible to integrate other motions. In particular, free motions that correspond to zero propulsion force trajectories are clearly of interest. The basic idea is to employ trajectory tracking controllers and positively invariant sets associated with the trajectory tracking error dynamics to define a notion of connectivity to and from such a trajectory. Note however, that several simplifications occur for static points that are not available for more general trajectories, and as such, the subsequent development only makes use of static points.

B. Multigain LQ Controller

A conventional LQ feedback

$$U = K(X - X_e(r)) + \Gamma r = KX + H(K)r \tag{6}$$

is used to control the spacecraft thrust to arrive at a commanded point (5), where

$$\Gamma = \begin{bmatrix} -3n^2 m_c & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & n^2 m_c \end{bmatrix}, \quad H(K) = \Gamma - K\begin{bmatrix} I_3 \\ 0_3 \end{bmatrix}$$

and where I_3 denotes the 3×3 identity matrix and 0_3 denotes the 3×3 zero matrix. The LQ controller provides an asymptotically stable closed-loop system but does not enforce obstacle avoidance constraints.

To provide greater flexibility in handling constraints, a multigain controller architecture is employed [15]. Specifically, it is assumed that a finite set of LQ gains $K \in \mathcal{K} = \{K_1, \cdots, K_m\}$ are available to control the spacecraft. Although feedback gains can be generated using other techniques, the use of LQ is practically convenient. By using a large control weight in the LQ cost functional, motions with low fuel consumption yet large excursions can be generated; using a large control weight in the LQ cost, motions with short transition time can be generated. See [33] for a detailed study of the dependence between LQ cost weights and fuel consumption metrics.

C. Positively Invariant Sets

An ellipsoidal set, characterized by a relative position $r \in \mathcal{N}$ and corresponding to a gain $K \in \mathcal{K}$, is defined by

$$\bar{C}(r,K) = \left\{ X \in R^6 \colon \frac{1}{2}(X - X_e(r))^T P(K)(X - X_e(r)) \le 1 \right\} \subset R^6 \tag{7}$$

where

$$\bar{A}(K)^T P \bar{A}(K) - P < 0 \tag{8}$$

$\bar{A}(K) = (A + BK)$ and $P = P(K) > 0$. This set is positively invariant for closed-loop dynamics. Positive invariance implies that any trajectory of the closed-loop system that starts in $\bar{C}(r, K)$ is guaranteed to stay in $\bar{C}(r, K)$ as long as the same gain K is used and the set-point command r is maintained.

Positively invariant sets are useful for constraint enforcement. If the constraints are satisfied pointwise in the positively invariant set (i.e., the set is not only positively invariant but is also safe), the trajectory starting in the set will satisfy the constraints for all future times assuming that the set-point and the gains do not change.

To achieve the positive invariance, the matrix P can be obtained as the solution of the discrete-time Riccati equation in the LQ problem or as the solution of the preceding Lyapunov equation for the closed-loop asymptotically stable system. Note that because the system is linear, the positive invariance of $\bar{C}(r,K)$ implies the positive invariance of the scaled set

$$C(r,K,\rho)=\left\{X\in R^6\colon \frac{1}{2}(X-X_e(r))^T P(K)(X-X_e(r))\leq\rho^2\right\},\quad \rho\geq 0$$

Geometrically, the set $C(r,K,\rho)$ corresponds to an ellipsoid scaled by the value of ρ and centered around $X_e(r)$, $r\in\mathcal{N}$. In the case of bounded disturbances, the sets $C(r,K,\rho)$ will be invariant for all sufficiently large ρ. See Sec. VI.

IV. Single Stationary Obstacle Avoidance

In this section, the virtual net approach is developed for the case of avoiding a single stationary obstacle. A stationary obstacle refers to an obstacle whose Hill's frame coordinates remain constant in time. Such an obstacle can correspond to an object in the same orbital track as the spacecraft, to a maneuvering obstacle that manages to maintain its relative position, or to an imagined obstacle that is used to cover a moving obstacle's path (see Sec. V). The underlying idea is to maintain the spacecraft within a tube formed by safe positively invariant sets that do not intersect with the obstacle.

Consider $r_i\in\mathcal{N}$, representing a possible position on the net that the spacecraft can move to as a part of the obstacle avoidance maneuver. Suppose $O\subset R^6$ corresponds to the relative positions and velocities occupied by the obstacle (see Sec. IV.A for the actual obstacle representation) and that the current state of the spacecraft is $X(t_0)$ at the time instant $t_0\in Z^+$. If there exists a $\rho\geq 0$ and $K_j\in\mathcal{K}$ such that

$$X(t_0)\in C(r_i,K_j,\rho)\quad\text{and}\quad O(q,Q)\cap C(r_i,K_j,\rho)=\varnothing \tag{9}$$

the spacecraft can move to the position $r_i\in\mathcal{N}$ by engaging the control law with $r(t)=r_i$ and $K(t)=K_j$, $t\geq t_0$, and without hitting the obstacle confined to O.

The details of the approach are now presented.

A. Obstacle Representation

The set $O(q,Q)$, centered around the position $q\in R^3$, is used to overbound the position of the obstacle, i.e.

$$O(q,Q)=\{X\in R^6\colon (SX-q)^T Q(SX-q)\leq 1\} \tag{10}$$

where $Q=Q^T>0$ and

$$S=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\end{bmatrix} \tag{11}$$

The set $O(q,Q)$ can account for the obstacle and spacecraft physical sizes and for the uncertainties in the estimation of the obstacle/spacecraft positions. Note that the set $O(q,Q)$ has an ellipsoidal shape in the position directions and is unbounded in the velocity directions. Ellipsoidal sets, rather than polyhedral sets, are used here to overbound the obstacle, because ellipsoidal bounds are typically produced by position estimation algorithms, such as the extended Kalman filter.

B. Growth Distances

Motivated by [30], the minimum value of $\rho\geq 0$ for which $O(q,Q)\cap C(r,K,\rho)\neq\varnothing$ is referred to as the growth distance. This growth distance can also be viewed as the least upper bound on the values of ρ for which $O(q,Q)$ and $C(r,K,\rho)$ do not intersect. In Fig. 2, the positively invariant set is grown until touching the obstacle. The spacecraft can then move from any of the points on the virtual net inside the positively invariant set $C(r,K,\rho)$ to $X_e(r_i)$ marked with ‘x’ without colliding with the obstacle. The notation $\rho_g(r,K,Q,q)$ is used to reflect the dependence of the growth distance on the set-point $r\in\mathcal{N}$, the control gain $K\in\mathcal{K}$, and the obstacle parameters Q and q.

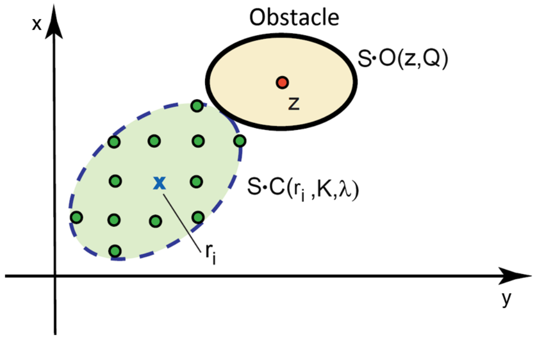


Fig. 2 Positively invariant set grown until touching obstacle.

Note that the growth distance depends on the position and shape of the obstacle (i.e., q and Q), which may be unknown in advance. Consequently, growth distance computations have to be performed online.

Because spacecraft have limited thrust, an additional maximum value of $\rho=\rho_u(r,K)$ has been defined, for which $X\in C(r,K,\rho_u(r,K))$ implies that the thrust $U=KX+H(K)r$ satisfies the imposed thrust limits. Here, ρ_u is referred to as the thrust limit on growth distance. Unlike ρ_g, the value of ρ_u does not depend on the position or shape of the obstacle and can be precomputed offline.

Finally, the thrust-limited growth distance is defined as

$$\rho^*(r,K,Q,q)=\min\{\rho_g(r,K,Q,q),\rho_u(r,K)\} \tag{12}$$

Note that $X(t_0)\in C(r_i,K_j,\rho^*(r_i,K_j,Q,q))$ implies that the ensuing closed-loop spacecraft trajectory under the control (6), where $r(t)=r_i$ and $K(t)=K_j$ for $t\geq t_0$, satisfies the thrust limits and avoids collisions with an obstacle confined to $O(q,Q)$.

The preceding definitions were given for the case of a single stationary obstacle $O(q,Q)$. In the case of multiple obstacles, the growth distance is replaced by the multigrowth distance, which is the minimum growth distance to each of $O(q_l,Q_l)$, $l=1,\cdots,n_d$.

C. Growth Distance Computations

Define $\bar{X}=X-X_e(r)$ and $\alpha=2\rho^2$. The problem of determining the growth distance $\rho_g(r,K,Q,q)$ reduces to the constrained optimization problem

$$\min_{\alpha,\bar{X}}\ \alpha$$

$$\text{subject to } \bar{X}^T P\bar{X}\leq\alpha$$

$$((S(\bar{X}+X_e(r))-q)^T Q((S(\bar{X}+X_e(r))-q)\leq 1 \tag{13}$$

where a minimum size invariant ellipsoid is found that shares a common point with the obstacle. To solve this optimization problem, the Karush–Kuhn–Tucker conditions are used ([34] pp. 342–345, [35] pp. 162–167). Note that standard linear independence constraint qualification conditions hold given that $P>0$. In addition,

$$\mathcal{L}=\alpha+\lambda_1(\bar{X}^T P\bar{X}-\alpha)$$
$$+\lambda_2((S(\bar{X}+X_e(r))-z)^T Q(S(\bar{X}+X_e(r))-q)-1)$$

where λ_1 and λ_2 are Lagrange multipliers. The stationarity of the Lagrangian (setting partial derivative equal to zero) with respect to α yields $\lambda_1=1$. The stationarity of the Lagrangian with respect to $\bar{X}$ yields

$$\bar{X}=\bar{X}(\lambda_2,r,q)=-(P+\lambda_2 S^T QS)^{-1}S^T Q(SX_e(r)-q)\lambda_2 \tag{14}$$

where the scalar $\lambda_2\geq 0$ is to be determined. Note that $P>0$, $S^T QS\geq 0$, $\lambda_2\geq 0$ (as the Lagrange multiplier corresponding to an inequality constraint) imply that $(P+\lambda_2 S^T QS)$ is invertible. The problem reduces to finding a nonnegative scalar λ_2, which is the root of

$$F(\lambda_2, r, q) = (SX - q)^T Q(SX - q) - 1 = 0 \tag{15}$$

where

$$X = \bar{X}(\lambda_2, r, q) + X_e(r)$$

The scalar root finding problem (15) has to be solved online multiple times for different $r \in \mathcal{N}$, and in the case of avoiding a predicted obstacle path also for different q. To quickly solve this problem, while reusing previously found solutions as approximations, a dynamic Newton–Raphson's algorithm is used ([35] pp. 308–309, [36]). This algorithm uses predictor–corrector updates to track the root as a function of q and r, and is given by

$$\begin{aligned}\lambda_2^{k+1,+} &= \lambda_2^k + \left\{\frac{\partial F}{\partial \lambda_2}(\lambda_2^k, q^k, r^k)\right\}^{-1}\Bigg\{-F(\lambda_2^k, q^k, r^k)\\ &\quad -\frac{\partial F}{\partial z}(\lambda_2^k, q^k, r^k)(q^{k+1} - q^k) - \frac{\partial F}{\partial r}(\lambda_2^k, q^k, r^k)(r^{k+1} - r^k)\Bigg\},\\ \lambda_2^{k+1} &= \max\{0, \lambda_2^{k+1,+}\}\end{aligned}$$

To implement the algorithm, the known functional form for F is taken advantage of and explicitly compute the partial derivatives

$$\begin{aligned}\frac{\partial \bar{X}}{\partial \lambda_2} &= (P + \lambda_2 S^T Q S)^{-1}\left\{-S^T Q(SX_e(r) - q) - S^T Q S\bar{X}\right\},\\ \frac{\partial F}{\partial \lambda_2} &= 2(SX - q)^T Q\left(S\frac{\partial \bar{X}}{\partial \lambda_2}\right),\\ \frac{\partial \bar{X}}{\partial r} &= (P + \lambda_2 S^T Q S)^{-1}\{-S^T Q S\Omega\}\lambda_2,\\ \frac{\partial F}{\partial r} &= 2(S\bar{X} - q + r)^T Q\left(S\frac{\partial \bar{X}}{\partial r} + I_3\right),\\ \frac{\partial \bar{X}}{\partial q} &= (P + \lambda_2 S^T Q S)^{-1} S^T Q S\Omega\lambda_2,\\ \frac{\partial F}{\partial q} &= 2(S\bar{X} - q + r)^T Q\left(S\frac{\partial \bar{X}}{\partial z} - I_3\right)\end{aligned} \tag{16}$$

where, $X_e(r) = \Omega r$

$$\Omega = \begin{bmatrix} I_3 \\ 0 \end{bmatrix}$$

and I_3 denotes the 3×3 identity matrix. Note that $S\Omega = I_3$.

Figure 3 illustrates growth distance tracking. For the first 20 iterations, r^k is held constant to enable initial convergence of the algorithm. Then, r^k varies through the virtual net. One iteration of the Newton–Raphson algorithm per value of r^k is used to update the root λ_2^{k+1}. Figure 3b demonstrates that the growth distance tracking is accurate. The growth distance is occasionally zero, indicating an overlap between several r^k and the obstacle. Figure 3c illustrates the trajectory of r^k in three dimensions.

D. Thrust Limit on Growth Distance Computations

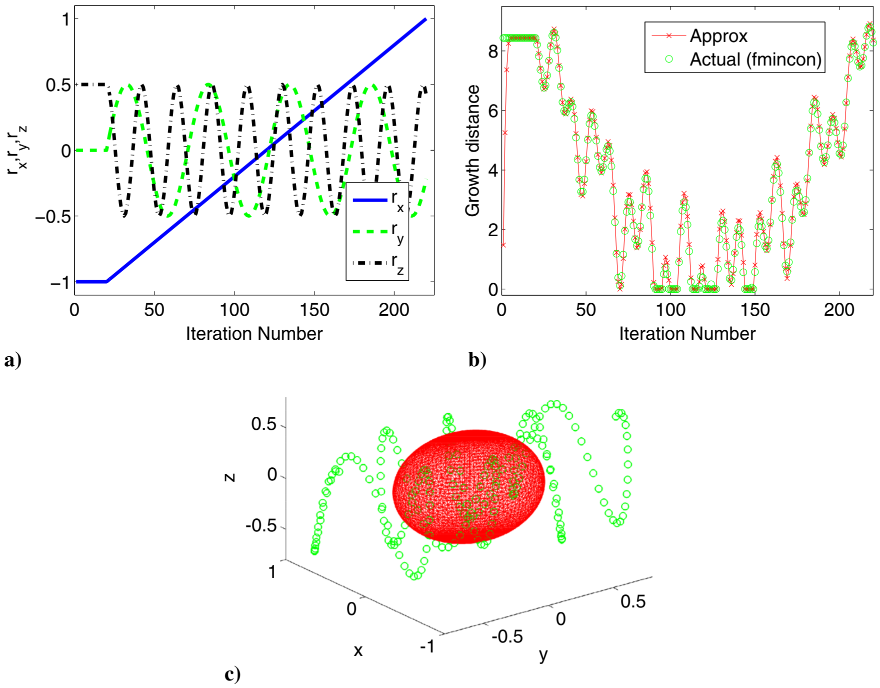
Suppose that the thrust limits are expressed in the form $\|LU\| \le 1$ for an appropriately defined matrix L and norm $\|\cdot\|$. The computational procedures to determine $\rho_u(r, K)$ involve solving a bilevel optimization problem where $\|L(KX + H(K)r)\|$ is maximized subject to the constraint $X \in C(r, K, \alpha)$, and bisections are performed on the value of α so that the maximum value is driven to one. As demonstrated in this section, in special cases, this computation can be greatly simplified.

Suppose that the thrust constraints are prescribed in terms of polyhedral norm bounds, specifically

$$e_i^T(KX + Hr) \le u_{\max}, \qquad i = 1, 2, \cdots, m, \tag{17}$$

where e_i denotes the vertices of the unit norm polytope and $u_{\max}$ is the norm bound. The infinity norm, for instance, has $m = 6$, and

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},$$
$$e_4 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, \quad e_5 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad e_6 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \tag{18}$$



a)

b)

c)

Fig. 3 Growth distance tracking. a) components of r, r_x, r_y, and r_z varying vs iteration number, b) growth distance vs iteration number computed by the dynamic Newton–Raphson algorithm, and c) trajectory of r and obstacle (ellipsoid).

In the case of nonpolyhedral norm bounds, such as the two-norm, an approximation by a polyhedral norm bound can be employed.

The thrust limit on the growth distance is then determined based on solving, for $i = 1, \cdots, n$, the optimization problems

$$\begin{aligned}&\text{maximize } e_i^T(KX + Hr)\\ &\text{subject to } \frac{1}{2}(X - X_e(r))^T P(X - X_e(r)) \leq c\end{aligned} \tag{19}$$

If the value of c is found for which the solutions X_i^* of Eq. (19) satisfy $\max_i\{e_i^T(KX_i^* + Hr)\} = u_{\max}$, then $\rho_u(r, K) = \sqrt{c}$.

The problem (19) can be solved by diagonalizing the matrix $P = P^T > 0$, using an orthogonal matrix, V

$$P = V^T \Lambda V, \qquad \Lambda = \mathrm{diag}[\lambda_1^2, \cdots, \lambda_6^2], \qquad \lambda_i > 0$$

By defining $z = X - X_e(r)$ and ζ so that

$$z = V^T \Lambda^{-\frac{1}{2}} \zeta$$

it follows that

$$z^T P z = \zeta^T \Lambda^{-\frac{1}{2}} V P V^T \Lambda^{-\frac{1}{2}} \zeta = \zeta^T \zeta$$

The problem (19) can now be rewritten as

$$\begin{aligned}&\text{maximize } h_i^T \zeta + e_i^T \Gamma r\\ &\text{subject to } \frac{1}{2}\zeta^T \zeta \leq c\end{aligned} \tag{20}$$

where

$$h_i^T = e_i^T K V^T \Lambda^{-\frac{1}{2}}$$

The solution to the constrained maximization problem (20) of maximizing the inner product of two vectors over a unit two-norm ball is given by

$$\zeta_i = \frac{h_i}{\|h_i\|}\sqrt{2c} \tag{21}$$

where $\|\cdot\|$ denotes the vector two-norm. The maximum value of the objective function in Eq. (19) is given by

$$\|h_i\|\sqrt{2c} + e_i^T \Gamma r$$

Consequently, to satisfy Eq. (17), let

$$c = \begin{cases} 0, & \text{if } \exists i\text{: } u_{\max} \leq e_i^T \Gamma r,\\ \min\limits_i \frac{1}{2}\left(\frac{u_{\max} - e_i^T \Gamma r}{\|h_i\|}\right)^2, & \text{otherwise}\end{cases} \tag{22}$$

Thus, the problem of finding the thrust limit on the growth distance for polyhedral norm bounds has an explicit solution given by Eq. (22). Even though the computation of thrust limits on the growth distance can be performed offline for the nominal operating conditions, fast computational procedures are beneficial in case of thruster failures, degradations, and restrictions on thrust directions (e.g., caused by the presence of other spacecraft nearby), all of which can lead to changing constraints on thrust that need to be handled online during spacecraft missions.

Note that the condition $u_{\max} \geq \max_i\{e_i^T \Gamma r\}$ is satisfied if the available thrust can maintain the point $X_e(r)$ in steady state. It is also noted that, based on the form of Γ, c is independent of r_y, which is the in-track component of the point in the virtual net. Hence, the computations of $\rho_u(r, K)$ need only be performed with $r_y = 0$.

When a spacecraft does not have independent thrusters in the x, y, and z directions, a two-norm thrust limit is more practical. Unfortunately Eq. (19) is, in general, a nonconvex problem. In this case, the two-norm bound can be approximated by a polyhedral norm bound (17), with the vertices e_i selected on the unit two-norm ball in R^3. Note that higher accuracy of this approximation requires a higher number of vertices in Eq. (17), which complicates Eq. (22).

E. Connectivity Graph and Graph Search

The notion of connectivity between two vertices of the virtual net, $r_i \in \mathcal{N}$ and $r_j \in \mathcal{N}$, is now introduced. The vertex r_i is connected to the vertex r_j if there exists a gain $K \in \mathcal{K}$ such that

$$X_e(r_i) \in \mathrm{int}\, C(r_j, K, \rho^*(r_j, K, Q, q)) \tag{23}$$

where int denotes the interior of a set. The connectivity implies that a spacecraft located close to a point corresponding to r_i can transition to a point $X_e(r_j)$ by using limited thrust and avoiding collision with the obstacle. Note that if r_i is connected to r_j, it does not imply that, in turn, r_j is connected to r_i. It is also noted that connectivity depends on the existence of an appropriate control gain from the set of gains $\mathcal{K}$ but the condition (23) does not need to hold for all gains.

Because more than one path that avoids the obstacles and satisfies thrust limits may exist, offline-constructed unconstrained cost of transition matrices are used to search for an efficient path. For instance, to produce time- or fuel-efficient paths, offline simulated transitions between all $r_i, r_j \in \mathcal{N}$ for each $K \in \mathcal{K}$ are performed, and the time and fuel necessary to reach a prescribed neighborhood of $X_e(r_j)$ starting from $X_e(r_i)$ are recorded. The results are merged into an unconstrained cost of transition matrix that stores the respective minimum value, while in parallel, the control selectivity matrix identifies which gain $K^*(r_i, r_j)$ produced said minimum. Note that constraints are not accounted for in these offline constructed transitions.

The online motion planning with obstacle avoidance is performed according to the following procedure:

Step 1: Determine the obstacle location and shape (i.e., q and Q).

Step 2: By using fast growth distance computations, determine the thrust-limited growth distance based on Eq. (12), with ρ_g computed online and ρ_u precomputed offline.

Step 3: Construct a graph connectivity matrix between all $r_i, r_j \in \mathcal{N}$. In the graph connectivity matrix, if two vertices are not connected, the corresponding matrix element is $+\infty$; if they are connected, the corresponding matrix element is 1. The graph connectivity matrix is multiplied elementwise by offline constructed unconstrained cost of transition matrix to produce a constrained cost of transition matrix.

Step 4: Perform graph search (using any standard graph search algorithm) to determine a sequence of connected vertices $r[k] \in \mathcal{N}$ such that $r[1]$ satisfies the initial constraints, $r[l_p]$ satisfies the final constraints, and the cumulative transition cost computed from the constrained cost of transition matrix is minimized.

After the path has been determined as a sequence of the set-points, the execution of the path proceeds by checking if the current state $X(t)$ is in the safe positively invariant set corresponding to the next reference r^+ and control gain $K^+ = K^*(r, r^+)$ in the sequence; if it is, then the controller switches to this reference and control gain

$$X(t) \in C(r^+, K^+, \rho^*(r^+, K^+, Q, q)) \to r(t) = r^+,\ K(t) = K^+ \tag{24}$$

The online tasks of growth distance computation and graph search are computationally tractable onboard a spacecraft, both of which run in $O(n)$ time, where n is the number of nodes in the virtual net. Although the growth distance optimization problem (13) is complex, the reduction to a scalar root finding problem (15) and use of Newton–Raphson's algorithm to track the root (Sec. IV.C) enables performance linear in the number of nodes. Note that this computation is further parallelizable, and that the determination of the other required parameters (P, ρ_u) is performed offline.

A variety of algorithms for graph search exist, such as A-star, Breadth-First-Search, and Dijkstra's algorithm, that have been used in navigation, routing systems, and autonomous vehicle applications with comparable onboard computing power. Note that due to the existence of thrust limits on growth distance, the connectivity graph is very sparse and, as such, the time complexity of search is linear in the number of nodes. Graph search algorithms have been previously researched in the context of spacecraft control (see [24–27]). Finally, note that the computation effort vs performance tradeoff can be managed by reducing the number of nodes in the virtual net.

V. Moving Obstacle Avoidance Approach

An obstacle is considered moving if its relative coordinates in Hill's frame change with time. There are two approaches to avoid moving obstacles. The first approach, suitable for faster moving obstacles, is to cover the obstacles' paths by a union of a finite number of ellipsoidal sets. Then, the problem reduces to that of avoiding multiple stationary obstacles and can be treated using a slight modification of the approach already presented in Sec. IV. Note, however, that this approach is conservative, as it does not account for the obstacles progressions along their paths vs time. It is appropriate for obstacles moving at high relative velocity as compared to spacecraft maneuvering authority. The second approach, which can be used in the case of slower moving obstacles, is to account for their motion in time, by expanding the vertices of the connectivity graph, previously based solely on N, with time information. To reduce conservatism in this second approach, the sets satisfy a contractivity property.

A. Avoiding Moving Obstacles by Covering their Paths with a Union of Ellipsoidal Sets

To avoid a nonstationary obstacle, its path can be covered by a union of a finite number of ellipsoidal sets

$$\mathcal{D}=\bigcup_{l=1}^{l=n_d} O(z_l,Q_l) \tag{25}$$

where the center of the lth set is denoted by $z_l \in R^3$ and the lth set shape is defined by $Q_l = Q_l^T > 0$. Then, the obstacle avoidance condition for the closed-loop trajectory that emanates from $X(0)$ with the set-point r_i and gain K_j is given by

$$\begin{gathered}X(0)\in C(r_i,K_j,\rho) \quad \text{and} \quad O(z_l,Q_l)\cap C(r_i,K_j,\rho)=\varnothing,\\ \text{for all } l=1,\cdots,n_d\end{gathered} \tag{26}$$

The same approach, with larger n_d, can be used to handle multiple moving obstacles.

B. Avoiding Moving Obstacles by Accounting for Their Motion in Time

To treat slower moving obstacles, the notion of time is introduced into the problem, whereas a transition between r_i and r_j might not be feasible if it is initiated at time t_1. However, based on the motion of obstacles, it might become feasible if it is initiated at time t_2. To accommodate moving obstacles, sets $C_k(r,K,\rho)$, $0\le k\le N$, satisfying the following contractivity property

$$\bar{A}(K)^k(C_k(r,K,\rho)-\{X_e(r)\})\subseteq(C(r,K,\rho)-\{X_e(r)\}) \tag{27}$$

are employed. Note that if $X(0)\in C_k(r,K,\rho)$, then $X(1)\in C_{k-1}(r,K,\rho)$, $X(2)\in C_{k-2}(r,K,\rho),\cdots,X(k)\in C_0(r,K,\rho)=C(r,K,\rho)$. The set $C_k(r,K,\rho)$ can be much larger than $C(r,K,\rho)$; any states in $C_k(r,K,\rho)$ contract to $C(r,K,\rho)$ in k steps.

The connectivity between two vertices of the virtual net $r_i\in\mathcal{N}$ and $r_j\in\mathcal{N}$ at a specified time t_0 is now defined. This notion is based on the fact that the time to transition from any state in $C_N(r,K,\rho)$ to $C(r,K,\rho)$ is less or equal than N steps. Suppose that the obstacle path $D(t_0\colon t_0+N\cdot H)$ has been predicted over the $N\cdot H$ discrete steps from the time instant t_0, where

$$D(t_k\colon t_r)=\bigcup_{t=t_k}^{t=t_r} O(z(t),Q(t))$$

The node $r_i\in\mathcal{N}$ is connected to $r_i\in\mathcal{N}$ at the time instant $t_k=t_0+kN$ if there exists $K\in\mathcal{K}$ such that

$$D(t_k\colon t_k+N)\cap C(r_i,K,\rho)=\varnothing \tag{28}$$

The node $r_i\in\mathcal{N}$ is connected to node $r_j\in\mathcal{N}$ at the time instant t_k if there exists $K\in\mathcal{K}$ such that

$$D(t_k\colon t_k+N)\cap C_N(r_j,K,\rho)=\varnothing \tag{29}$$

and

$$C(r_i,K,\rho)\subset C_N(r_j,K,\rho) \tag{30}$$

The connectivity implies that a spacecraft located close to a point corresponding to r_i, $X_e(r_i)$ can transition close to a point $X_e(r_j)$ between the time instants t_k and t_k+N while avoiding collision with the obstacle. Note that if r_i is connected to r_j, it does not imply that, in turn, r_j is connected to r_i, and that connectivity depends on the existence of an appropriate control gain from the set of gains $\mathcal{K}$ but does not need to hold for all gains. Furthermore, because connectivity depends on the predicted motion of the obstacle, connectivity/nonconnectivity can depend on time.

The online motion planning with obstacle avoidance is performed according to the following procedure:

Step 1: Determine the obstacle location and shape and predict the obstacle path $D(t_0\colon t_0+N\cdot H)$.

Step 2: Construct graph connectivity matrices corresponding to t_k, $k=0,1,\cdots,H$. In the graph connectivity matrix, if two vertices r_i and r_j are not connected at t_k, the corresponding matrix element is $+\infty$; if they are connected, the corresponding matrix element is the cost of the transition. In parallel, build the control gain selectivity matrix, which identifies the index of the highest preference gain K for which r_i and r_j are connected. This gain will be applied if the edge connecting r_i and r_j is traversed.

Step 3: Perform graph search to determine a sequence $r[t_k]\in\mathcal{N}$ and control gains $K[k]\in\mathcal{K}$, $k=1,\cdots,l_p$, such that $r[t_1]$ satisfies the initial constraints, $r[l_p]$ satisfies the final constraints, and the cost function (such as the path length, expected fuel consumption, or expected maneuver time) is minimized.

Note a few facts. First, condition (29) is conservative. It can be replaced by a less conservative condition

$$D(t_k\colon t_k+m)\cap C_{N-m}(r_j,K,\rho)=\{\varnothing\},\quad m=0,1,\ldots,N$$

at the cost of more intensive computations. Second, the condition (29) is checked computationally using the fast growth distance algorithm described in Sec. IV.C. The intersection is empty if C_N can be grown prior to touching $D(t_k\colon t_k+N)$. This fast growth distance algorithm is essential to rapidly construct the connectivity matrices. Finally, in the subsequent simulations, the graph search is applied to a lifted graph, the vertices of which are the pairs (r_i,t_k). Note that for the moving obstacle case, graph-based techniques have an important advantage in that the computational time and effort associated with them increases linearly with time horizon.

VI. Bounded Disturbances

The obstacle avoidance approach can be easily extended to handle bounded disturbances due to spacecraft relative motion perturbations, including the effects of J_2 ([32] p. 219, [37] pp. 85–87]), nonzero eccentricity ([32] p. 104), thrust errors ([37] pp. 93–97), air drag ([37] p. 81), solar pressure ([37] pp. 87–89), rotational-translational kinematic coupling ([32] pp. 227–228), etc.

First, the case of multiple stationary obstacles is considered, where the system model (3) is extended with the disturbance input

$$X(t+1)=AX(t)+BU(t)+Bw(t) \tag{31}$$

where $w(t) \in W$, W is the convex hull of w^i for $i = 1, \ldots, n_w$, w^i denotes the vertices of a disturbance set, and n_w is the number of vertices. Note that W is a compact set.

The positive invariance of $C(r, K, \gamma)$, $\gamma > 0$, for $W = \{0\}$ has already been established. When $W \neq \{0\}$, it can be shown by employing an input-to-state stable-Lyapunov function

$$V = \frac{1}{2}(X - X_e(r))^T P(X - X_e(r)) \tag{32}$$

and by the compactness of W, that there exists $\gamma_{\min}$ such that $C(r, K, \gamma)$ is positively invariant for all $\gamma \geq \gamma_{\min}$. Note that $\gamma_{\min} = \gamma_{\min}(K)$.

Because $C(r, K, \gamma_{\min}(K))$ is disturbance invariant, it contains the minimum invariant set that is an attractor for closed-loop trajectories, as long as r and K are maintained at constant values. Hence, in the case of bounded disturbances, connectivity can be redefined by replacing $X_e(r_i)$ in Eq. (23) with $C(r_i, K, \gamma_{\min}(K))$. Specifically, the vertex $r_i \in \mathcal{N}$ is connected to the vertex $r_j \in \mathcal{N}$ if there exists $K \in \mathcal{K}$ such that

$$C(r_i, K, \gamma_{\min}(K)) \subset \operatorname{int} C(r_j, K, \rho^*(r_j, K, Q, q)) \tag{33}$$

The condition (33) ensures that a switch from r_i to r_j may occur and that subsequent dynamics will not lead to collision with the obstacle once $X(t) \in C(r_i, K, \gamma_{\min}(K))$.

To compute $\gamma_{\min}$ under all possible $w \in W$, it is sufficient to examine the flow at the vertices w^i of the disturbance set and demonstrate that if $X_k \in C(r, K, \gamma(K))$ and $w \in \{w^i, i = 1, \ldots, n_w\}$, then $X_{k+1} \in C(r, K, \gamma(K))$. The value $\gamma_{\min}$ is the minimum γ for which this condition holds.

To find $\gamma_{\min}$, a bilevel optimization strategy is used, where the inner loop solves n_w optimization problems numerically with respect to X

$$\begin{aligned} &\text{maximize } F_i(X) = \frac{1}{2}(AX + BU + Bw^i)^T \frac{P(K)}{\gamma_i^2}(AX + BU + Bw^i), \\ &\text{subject to } \frac{1}{2}(X - X_e(r))^T P(K)(X - X_e(r)) \leq \gamma_i^2 \end{aligned} \tag{34}$$

and the outer loop performs bisections on each γ_i, so that all $F_i(X^*(\gamma_i))$, where $X^*(\gamma_i)$ denotes the inner-loop solution, are driven to 1. Thus, $\gamma_{\min} = \min(\gamma_i)$ for $i = 1, \ldots, n_w$. Note that $\gamma_{\min}$ is independent of position r, and so this calculation may be done once offline for each $K \in \mathcal{K}$ and stored onboard for real-time implementation.

The approach of Sec. V for handling moving obstacles while accounting for their motion in time also admits a straightforward extension to the bounded disturbance case. The main change is replacing condition (27) by

$$\begin{aligned} &\bar{A}(K)(C_k(r, K, \rho) - \{X_e(r)\}) \bigoplus BW \subseteq (C_{k-1}(r, K, \rho) - \{X_e(r)\}), \\ &C_0(r, K, \rho) = C(r, K, \rho) \end{aligned} \tag{35}$$

where $\bigoplus$ denotes the Minkowski sum ([14], [16] p. 76).

VII. Simulation Results

Simulations are now provided to illustrate these obstacle avoidance approaches. For these simulations, consider a nominal circular orbit of 850 km and discretize the CWH equations with a sampling period ΔT of 30 s. A virtual net covering approximately a 2 km cube is constructed. The spacecraft mass is $m = 140$ kg and $\mathcal{K} = \{K_1, K_2, K_3\}$, where K_1, K_2, K_3 are the LQ gains associated with state and control weight matrices $Q_{\text{lqr}} = \text{diag}(100, 100, 100, 10^7, 10^7, 10^7)$, and $R_{\text{lqr},1} = 2 \times 10^5 I_3$, $R_{\text{lqr},2} = 2 \times 10^7 I_3$, and $R_{\text{lqr},3} = 2 \times 10^9 I_3$. These gains are chosen to represent preferences for fuel considerations K_3, maneuver time considerations K_1, and a compromise between them K_2. A maximum thrust constraint of 5 N in each axis is also imposed. In all simulations, Dijkstra's algorithm is used to find the shortest cost path from initial node to final node.

A. Stationary Obstacles

Consider an ellipsoidal set $O(z_1, Q_1)$ overbounding an obstacle centered at $z_1 = [0.3 \quad 0.5 \quad 0.5]^T$ km, where $Q_1 = 100 I_3$. The spacecraft's initial condition is $X(0) = X_e(r_0)$, where $r_0 = [0.32 \quad 01.61]^T$ km, whereas the target's equilibrium node is $X_e(0)$. In the first example, cost of transition between any two nodes in the virtual net is set to 1 if the nodes are connected and $+\infty$ if not. Using the stationary obstacle avoidance approach presented in Sec. IV and the gain K_2 results in the closed-loop trajectory shown in Fig. 4. In Fig. 4a, the "x's" correspond to the initial and final nodes, the solid ellipsoid represents the obstacle, the solid curve is the spacecraft trajectory, and the transparent ellipsoids correspond to the safe positively invariant sets along the path. Figure 4b shows the magnitude of the spacecraft's thrust as a function of time. Note that the positively invariant sets are grown to different sizes, depending on both the thrust limit and the constraint that the sets must not intersect with the obstacle. Using these sets, the spacecraft is able to complete the desired avoidance maneuver well within maximum thrust constraints. The simulation is rerun for a grid of initial conditions and the resultant trajectories are shown in Fig. 5. Each x marks a different initial condition, and the solid curves represent the individual paths. The invariant set ellipsoids are omitted for visual clarity.

Next, a second obstacle $O(z_2, Q_2)$ centered at $z_2 = [0.3 \quad -0.4 \quad 0.5]^T$ is added, where $Q_2 = 100 I_3$. In calculating the growth distance, the minimum distance to each of $O(z_i, Q_i)$, $i = 1, 2$ is taken. Figure 6 displays the closed-loop path of the spacecraft and the time history of thrust magnitude.

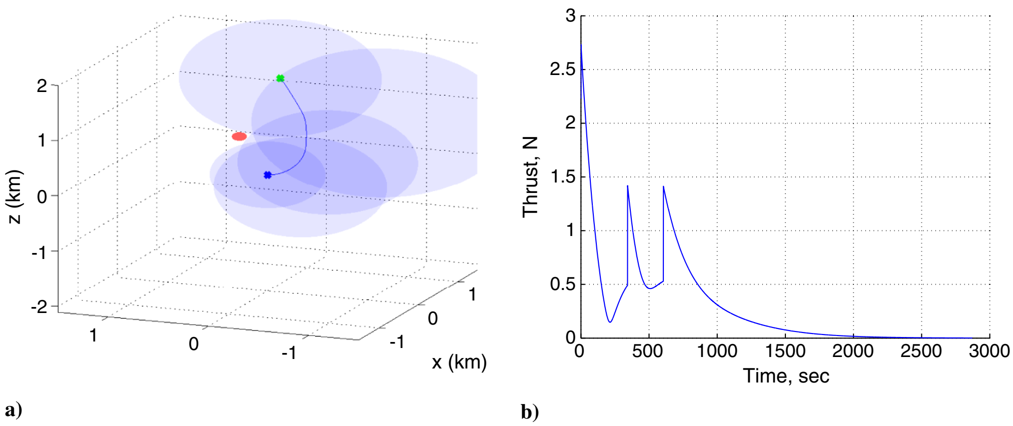


Fig. 4 Obstacle avoidance with a single stationary obstacle. a) Obstacle avoidance path and b) time history of thrust magnitude.

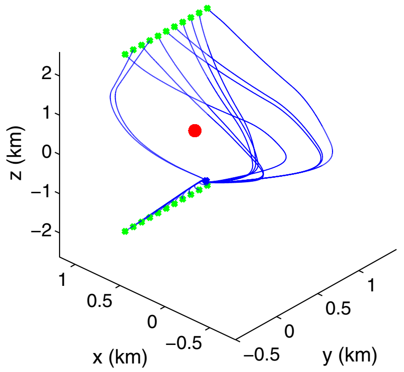
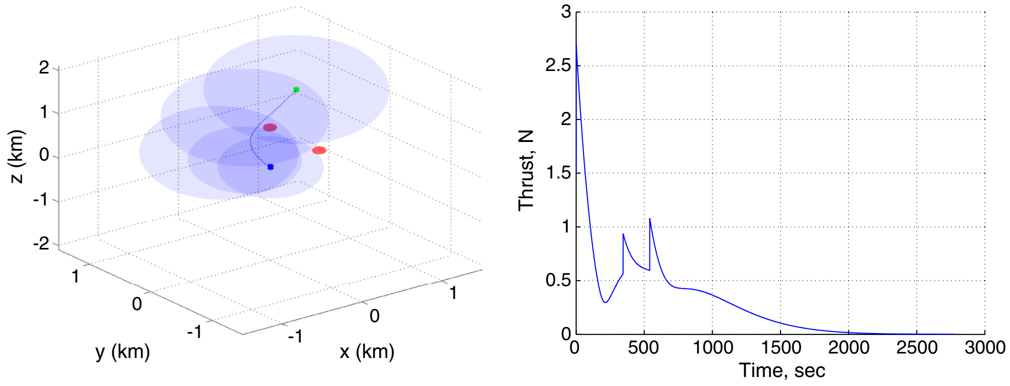


Fig. 5 Obstacle avoidance paths for many initial conditions.

B. Moving Obstacles

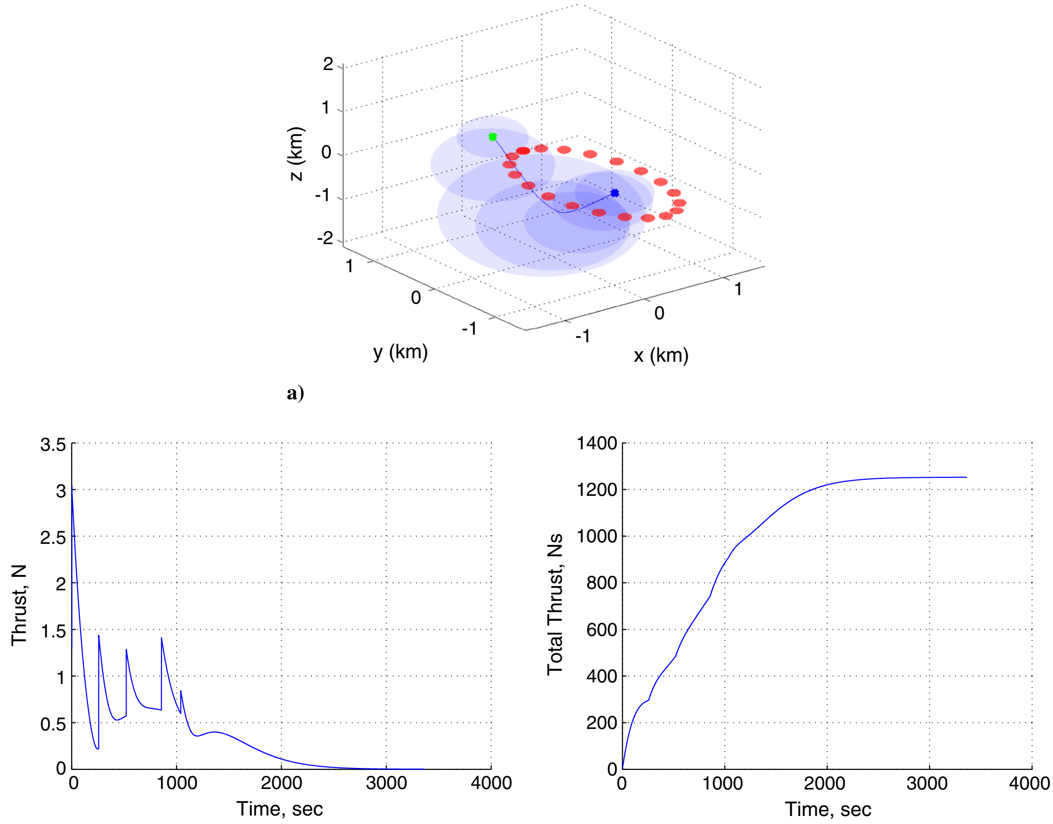
Obstacles in Hill's frame often have a natural, periodic motion. To account for this motion, consider the case of a nonstationary obstacle where its path is covered by the union of stationary obstacles. A union of ellipsoidal sets overbounds the obstacle' motion, where the obstacle's positions z_i are generated by sampling the relative motion of the obstacle with the initial condition $[0 \quad 0.5 \quad 0 \quad 0 \quad 0.006 \quad 0]^T$, and where $Q_i = 200I_3$, $i = 1 \ldots n_d$. The spacecraft's initial condition is $X(0) = X_e(r_0)$, where $r_0 = [0 \quad 1 \quad 0]^T$ km. The target equilibrium node is $X_e(r_d)$, where $r_d = [0 \quad -1 \quad 0]^T$ km. Figure 7 demonstrates that the spacecraft is able to avoid the closed obstacle path by hopping under it.

Cost matrices that reflect time and fuel costs of transition between nodes are considered next. The graph connectivity matrix is multiplied elementwise by these offline constructed unconstrained cost of transition matrices to produce constrained cost of transition matrices



a) b)

Fig. 6 Obstacle avoidance with two stationary obstacles. a) Obstacle avoidance path and b) time history of thrust magnitude.



a)

b) c)

Fig. 7 Nonstationary obstacle avoidance using the stationary obstacle approach. a) Obstacle avoidance path, b) time history of thrust magnitude, and c) cumulative thrust vs time.

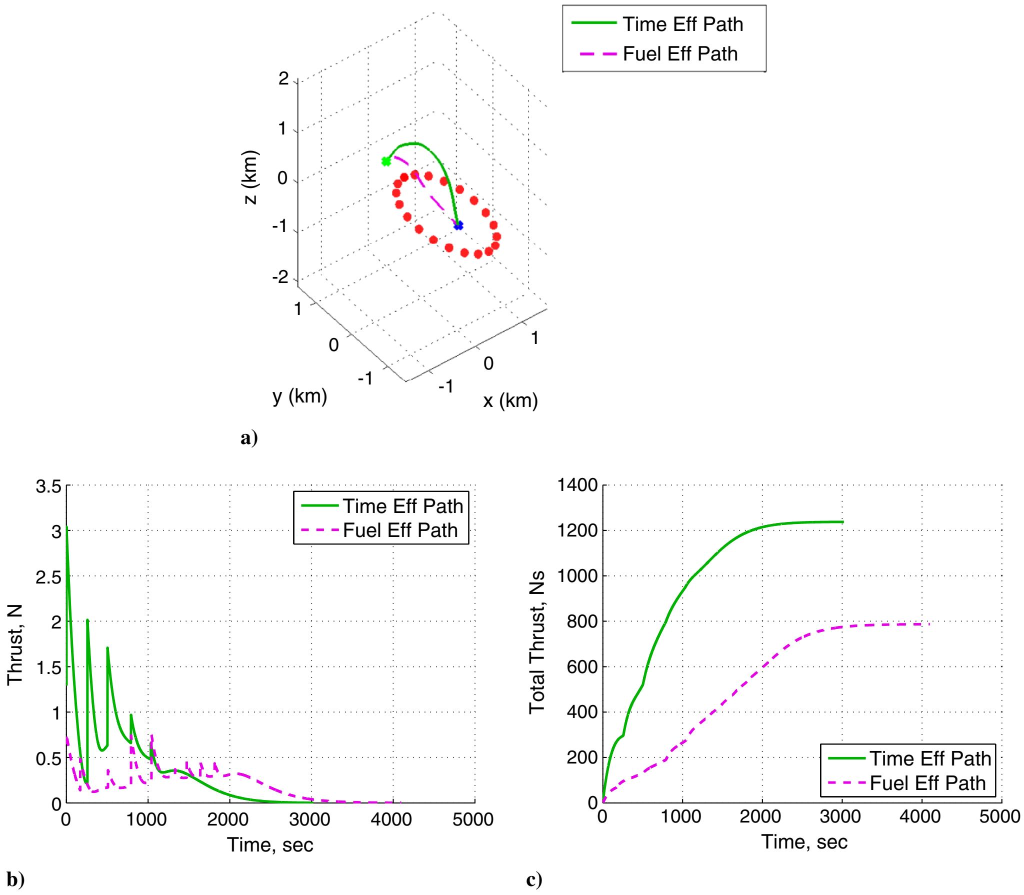


Fig. 8 Nonstationary obstacle avoidance using the stationary obstacle approach and cost matrices. a) Obstacle avoidance path, b) time history of thrust magnitude, and c) cumulative thrust vs time.

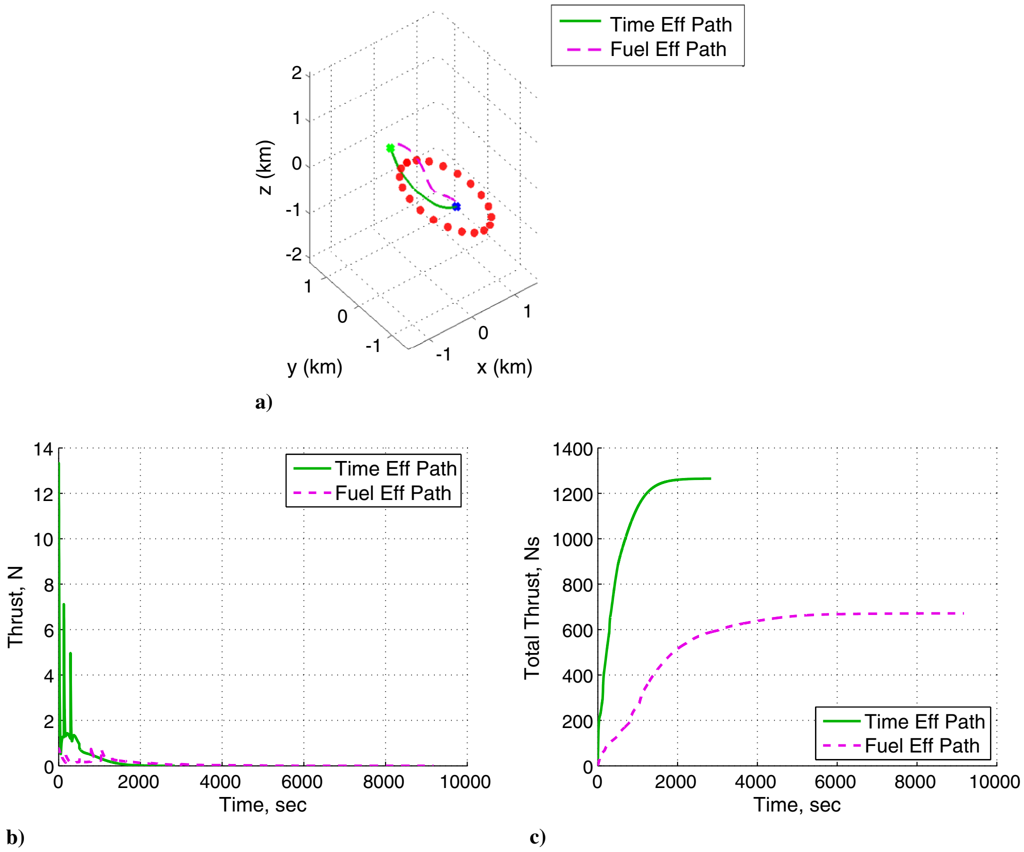


Fig. 9 Nonstationary obstacle avoidance using the stationary obstacle approach and gain switching. a) Obstacle avoidance path, b) time history of thrust magnitude, and c) cumulative thrust vs time.

Table 1 Total time and thrust for stationary obstacle approach using gain K_2

Path	Total time	Total thrust
Time-efficient path	3021.50 s	1237.620 N · s
Thrust-efficient path	4098 s	787.078 N · s

Table 2 Total time and thrust for stationary obstacle approach using the gain switching method

Path	Total time	Total thrust	Gains used
Time-efficient path	2841 s	1264.950 N · s	K_1, K_2
Thrust-efficient path	9177 s	671.297 N · s	K_2, K_3

for both time and fuel preferenced maneuvers. The simulation is repeated with these cost matrices and the resultant maneuvers are shown in Fig. 8. The solid line represents the time-efficient path, whereas the dashed line represents the fuel-efficient path. In comparing Figs. 7 and 8, it can be seen that using the cost matrices can help aid in finding fuel and time-efficient solutions. In Fig. 9, maneuvers are shown that use cost matrices that have access to all gains $K \in \mathcal{K}$, employing a multigain switching control architecture in order to produce more efficient maneuvers (see Tables 1 and 2).

The simulations are now repeated, taking into account the obstacle motion as a function of time using the approach presented in Sec. V. The spacecraft uses the single gain K_2 and uses the appropriate cost matrices to search for time- and fuel-efficient trajectories. Figure 10 shows that the graph search algorithm is able to find a trajectory that crosses the path of the obstacle while avoiding collision. The spacecraft waits for the obstacle to move, only then safely proceeding to the commanded node. Table 3 summarizes the maneuver time and thrust.

Table 3 Total time and thrust for moving obstacle approach using gain K_2

Path	Total time	Total thrust
Time-efficient path	5016 s	776.723 N · s
Thrust-efficient path	8355.5 s	745.57 N · s

Table 4 Total time and fuel for time-efficient trajectory comparison using gain K_1, stationary obstacle approach vs moving obstacle approach

Obstacles	Total time	Total thrust
Union of static obstacles	2655.0 s	1337.200 N · s
Moving obstacles	2574.5 s	1244.040 N · s

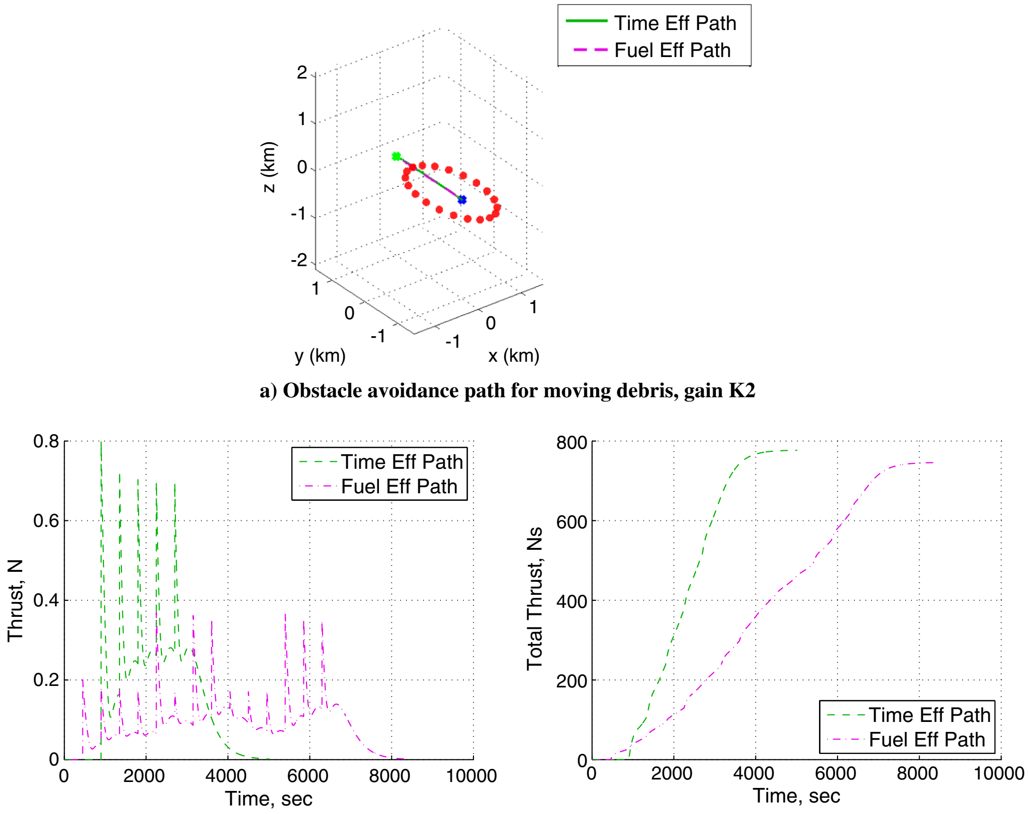
Table 5 Total time and fuel for time-efficient trajectory comparison using gain K_3, stationary obstacle approach vs moving obstacle approach

Obstacle	Total time	Total thrust
Union of static obstacles	11276 s	362.967 N · s
Moving obstacles	18875.5 s	272.722 N · s

The moving obstacle approach is now compared to the union of stationary obstacles approach. Gain K_1 is used when searching for a time-efficient solution and gain K_3 is used when searching for a fuel-efficient solution. Tables 4 and 5 compare time- and fuel-efficient maneuvers, respectively. The moving obstacle approach yields better time- and fuel-efficient solutions than the stationary obstacle approach at the cost of increased computational complexity.

C. Bounded Disturbances

Finally, the simulations are run for the case when bounded disturbances are present and the moving obstacle's path is covered by

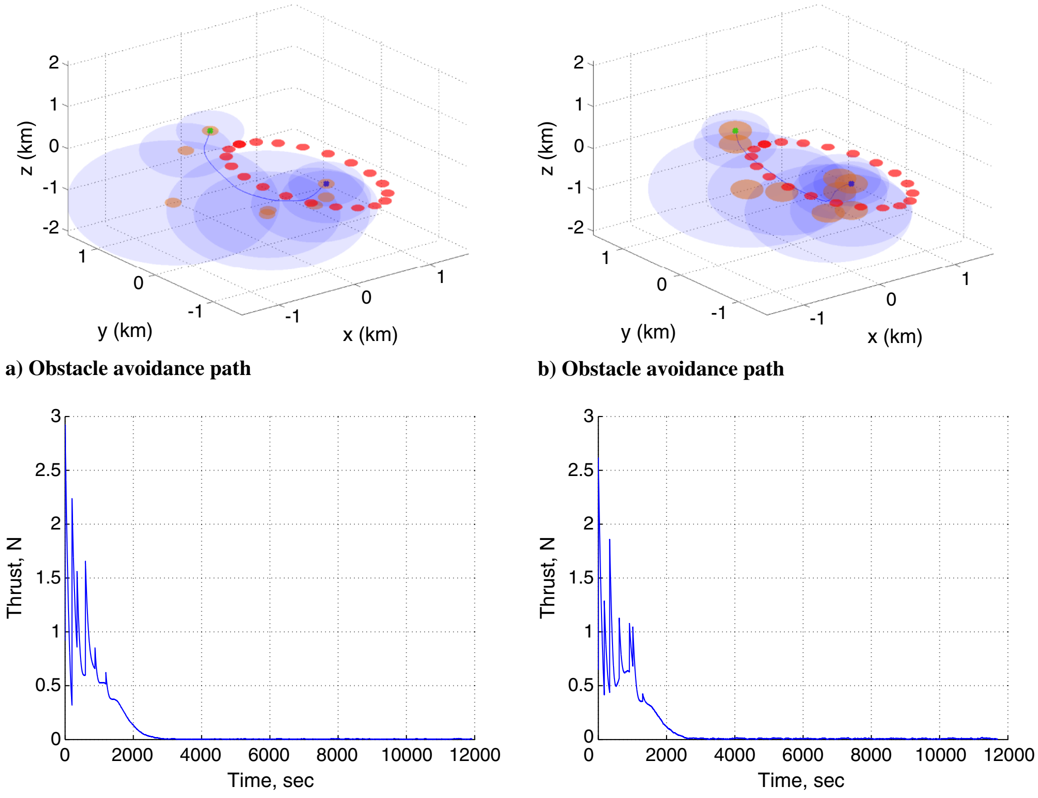


a) Obstacle avoidance path for moving debris, gain K2

b) Time history of thrust magnitude

c) Cumulative thrust

Fig. 10 Nonstationary obstacle avoidance using the moving obstacle approach and cost matrices. a) Obstacle avoidance path, b) time history of thrust magnitude, and c) cumulative thrust vs time.



a) Obstacle avoidance path

b) Obstacle avoidance path

c) Time history of thrust magnitude

d) Time history of thrust magnitude

Fig. 11 Nonstationary obstacle avoidance under uniform random disturbance. a) Obstacle avoidance path with $\varepsilon = 0.1$ N, b) obstacle avoidance path with $\varepsilon = 0.2$ N, and c) and d) time histories of thrust magnitude.

a union of stationary obstacles. It is considered that $W = \{w\colon \|w\|_\infty \leq \varepsilon\}$ for which $n_w = 8$, that is, disturbances that fit in a box of magnitude ε. Using the approach presented in Sec. VI, the gain K_2, and a uniform distribution of disturbances, for $\varepsilon = 0.1$ N and $\varepsilon = 0.2$ N, result in the closed-loop trajectories shown in Fig. 11. In this figure, the large transparent ellipsoids represent the maximally grown invariant sets $C(r, K, \rho^*(r, K, z))$ along the path and the corresponding smaller ellipsoids represent the disturbance invariant sets $C(r, K, \gamma_{\min}(K))$ along the path. The spacecraft is able to safely avoid the obstacle's path despite being subjected to disturbances.

VIII. Conclusions

By exploiting the properties of chained positively invariant sets around points in a virtual net, the problem of spacecraft constrained maneuver planning can be reduced to a graph search. Graph connectivity can be determined online based on thrust-limited growth distance computations. The application of a predictor-corrector Newton–Raphson type algorithm yields a fast procedure to determine the connectivity. With the proposed methodology, stationary and moving obstacles can be avoided, thrust limits can be enforced, and bounded additive unmeasured disturbances and uncertainties can be rigorously accounted for. By using transition cost matrices that emphasize fuel consumption or transition time, the developed framework effects maneuvers that consume less fuel or require less time. A moving obstacle can be avoided by accounting for its motion in time and exploiting contractive sets, or by avoiding the moving obstacle's path covered by a union of ellipsoidal sets. The former approach is less conservative than the latter and leads to improvements in time and fuel efficiency of the maneuvers at the cost of increased computation. The developed methodology generates feasible maneuver solutions to a difficult control problem with nonconvex, time-varying constraints, and unmeasured disturbances.

Acknowledgment

This work was supported by the U.S. Air Force Research Laboratory FA9453-12-1-0134.

References

[1] Committee for the Assessment of NASA's Orbital Debris Programs; National Research Council, Editor, *Limiting Future Collision Risk to Spacecraft: An Assessment of NASA's Meteoroid and Orbital Debris Programs*, The National Academies Press, Washington, D.C., 2011, p. 58.

[2] LaValle, S. M., *Planning Algorithms*, Cambridge Univ. Press, New York, 2006, p. 66.

[3] Maia, M. H., and Galvao, R. K. H., "On the Use of Mixed-Integer Linear Programming for Predictive Control with Avoidance Constraints," *International Journal of Robust and Nonlinear Control*, Vol. 19, No. 7, 2008, pp. 822–828.
doi:10.1002/rnc.1341

[4] Ranieri, C., "Path-Constrained Trajectory Optimization for Proximity Operations," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA, Reston, VA, Aug. 2008, pp. 1–17; also AIAA Paper 2008-6275.

[5] Betts, J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*, Society for Industrial and Applied Mathematics, Philadelphia, 2001, pp. 177–180.

[6] Epenoy, R., "Fuel Optimization for Continuous-Thrust Orbital Rendezvous with Collision Avoidance Constraint," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 2, 2011, pp. 493–503.
doi:10.2514/1.50996

[7] Martinson, N., "Obstacle Avoidance Guidance and Control Algorithms for Spacecraft Maneuvers," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA, Reston, VA, Aug. 2009, pp. 1–9; also AIAA Paper 2009-5672.

[8] Sanchez-Ortiz, N., Bello-Mora, M., and Klinkrad, H., "Collision Avoidance Manoeuvres During Spacecraft Mission Lifetime: Risk Reduction and Required ΔV," *Advances in Space Research*, Vol. 38, No. 9, 2006, pp. 2107–2116.
doi:10.1016/j.asr.2005.07.054

[9] Martinson, N., Munoz, J., and Wiens, G., "A New Method of Guidance Control for Autonomous Rendezvous in a Cluttered Space Environment," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA, Reston, VA, Aug. 2007, pp. 1–9; also AIAA Paper 2007-6351.

[10] Mueller, J. B., Griesemer, P. R., and Thomas, S., "Avoidance Maneuver Planning Incorporating Station-Keeping Constraints and Automatic Relaxation," *AIAA Infotech@Aerospace 2010*, AIAA, Reston, VA, April 2010, pp. 1–16; also AIAA Paper 2010-3525.

[11] Ulybyshev, Y., "Trajectory Optimization for Spacecraft Proximity Operations with Constraints," *AIAA Guidance, Navigation, and Control Conference*, AIAA, Reston, VA, Aug. 2011, pp. 1–19; also AIAA Paper 2011-6629.

[12] 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, Aug. 2002, pp. 755–764.
doi:10.2514/2.4943

[13] Mueller, J. B., "Onboard Planning of Collision Avoidance Maneuvers Using Robust Optimization," *AIAA Infotech@Aerospace Conference*, AIAA, Reston, VA, April 2009, pp. 1–17; also AIAA Paper 2009-2051.

[14] Kolmanovsky, I., and Gilbert, E., "Theory and Computation of Disturbance Invariant Sets for Discrete-Time Linear Systems," *Mathematical Problems in Engineering*, Vol. 4, No. 4, 1998, pp. 317–367.
doi:10.1155/S1024123X98000866

[15] Kolmanovsky, I., and Gilbert, E., "Multimode Regulators for Systems with State and Control Constraints and Disturbance Inputs," *Proceedings of Workshop Control Using Logic-Based Switching*, edited by Morse, A. S., Springer–Verlag, New York, 1997, pp. 104–117.

[16] Blanchini, F., and Miani, S., *Set-Theoretic Methods in Control*, Birkhäuser Boston, Cambridge, MA, 2007, pp. 110, 271–335.

[17] Blanchini, F., "Set Invariance in Control," *Automatica*, Vol. 35, No. 11, 1999, pp. 1747–1767.
doi:10.1016/S0005-1098(99)00113-2

[18] Wie, B., *Space Vehicle Dynamics and Control*, 2nd ed., AIAA, Reston, VA, 2008, p. 299.

[19] Blanchini, F., Pellegrino, F., and Visentini, L., "Control of Manipulators in a Constrained Workspace by Means of Linked Invariant Sets," *International Journal of Robust and Nonlinear Control*, Vol. 14, Nos. 13–14, 2004, pp. 1185–1205.
doi:10.1002/rnc.939

[20] Kolmanovsky, I., and Sun, J., "A Multi-Mode Switching-Based Command Tracking in Network-Controlled Systems with Pointwise-in-Time Constraints and Disturbance Inputs," *Proceedings of the 6th World Congress on Control and Automation*, IEEE, New York, 2006, pp. 199–204.

[21] Mayne, D., Seron, M., and Rakovic, S., "Robust Model Predictive Control of Constrained Linear Systems with Bounded Disturbances," *Automatica*, Vol. 41, No. 2, 2005, pp. 219–224.
doi:10.1016/j.automatica.2004.08.019

[22] Chisci, L., Rossiter, J., and Zappa, G., "Systems with Persistent Disturbances: Predictive Control with Restricted Constraints," *Automatica*, Vol. 37, No. 7, 2001, pp. 1019–1028.
doi:10.1016/S0005-1098(01)00051-6

[23] Ghaemi, R., Sun, J., and Kolmanovsky, I., "Robust Control of Constrained Linear Systems with Bounded Disturbances," *IEEE Transactions on Automatic Control*, Vol. 57, No. 10, 2012, pp. 2683–2688.
doi:10.1109/TAC.2012.2192362

[24] Mesbahi, M., and Hadaegh, F. Y., "Formation Flying Control of Multiple Spacecraft via Graphs, Matrix Inequalities, and Switching," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 369–377.
doi:10.2514/2.4721

[25] Choset, H., and Kortenkamp, D., "Path Planning and Control for Free-Flying Inspection Robot in Space," *Journal of Aerospace Engineering*, Vol. 12, No. 2, 1999, pp. 74–81.
doi:10.1061/(ASCE)0893-1321(1999)12:2(74)

[26] Guettier, C., and Poncet, J. C., "Multi-Levels Planning for Spacecraft Autonomy," *6th International Symposium on Artificial Intelligence and Robotics & Automation in Space* [CD-ROM], Canadian Space Agency, St-Hubert, Quebec, Canada, June 2001, pp. 1–8, available online at http://www.esa.int/TEC/Robotics/SEMMBJC4VUE_0.html.

[27] Frazzoli, E., "Quasi-Random Algorithms for Real-Time Spacecraft Motion Planning and Coordination," *Acta Astronautica*, Vol. 53, Nos. 4–10, 2003, pp. 485–495.
doi:10.1016/S0094-5765(03)80009-7

[28] Weiss, A., Baldwin, M., Petersen, C., Erwin, R. S., and Kolmanovsky, I., "Spacecraft Constrained Maneuver Planning for Moving Debris Avoidance Using Positively Invariant Constraint Admissible Sets," *Proceedings of the American Control Conference*, IEEE, New York, June 2013, pp. 4802–4807.

[29] Baldwin, M., Weiss, A., Kolmanovsky, I., and Erwin, R. S., "Spacecraft Debris Avoidance using Constraint Admissible Positively Invariant Sets," *Proceedings of AAS Space Flight Mechanics Meeting*, AAS Paper 2012-250, Univelt, Inc., Escondido, CA, 2012.

[30] Ong, C. J., and Gilbert, E. G., "Growth Distances: New Measures for Object Separation and Penetration," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 6, 1996, pp. 888–903.
doi:10.1109/70.544772

[31] Clohessy, W., and Wiltshire, R., "Terminal Guidance System for Satellite Rendezvous," *Aerospace Science and Technology*, Vol. 27, No. 9, 1960, pp. 653–658.
doi:10.2514/8.8704

[32] Alfriend, K. T., Vadali, S. R., Gurfil, P., How, J. P., and Breger, L. S., *Spacecraft Formation Flying*, Elsevier Astrodynamics Series, Oxford, 2010, pp. 219–221, 271–279.

[33] Park, H., Di Cairano, S., and Kolmanovsky, I., "Linear Quadratic Model Predictive Control Approach to Spacecraft Rendezvous and Docking," *Proceedings of 21st AAS/AIAA Space Flight Mechanics Meeting*, AAS Paper 2011-142, Univelt, Inc., Escondido, CA, 2011.

[34] Luenberger, D. G., and Ye, Y., *Linear and Nonlinear Programming*, Vol. 116, Springer, New York, 2008, pp. 342–345.

[35] Bazaraa, M. S., Sherali, H. D., and Shetty, C. M., *Nonlinear Programming: Theory and Algorithms*, Wiley-Interscience, New York, 2006, pp. 162–167.

[36] Quoc, T. D., Savorgnan, C., and Diehl, M., "Adjoint-Based Predictor-Corrector Sequential Convex Programming for Parametric Nonlinear Optimization," *SIAM Journal on Optimization*, Vol. 22, No. 4, 2012, pp. 1258–1284.
doi:10.1137/110844349

[37] Fehse, W., *Automated Rendezvous and Docking of Spacecraft*, Cambridge University Press, New York, 2003, pp. 85–87.