Real-Time Algorithm for Globally Optimal Impulsive Control of Linear Time-Variant Systems

Adam W. Koenig, Simone D’Amico
Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, USA

Abstract
This paper addresses the problem of minimizing the cost of a set of impulsive control inputs subject to the constraint that a linear time-variant dynamical system reaches a desired state at a specified time. Specifically, necessary and sufficient optimality conditions are derived for a broad class of optimal control problems where the cost of a control input is a time-varying norm-like function. This derivation uses reachable set theory to provide a simple geometric interpretation of the optimality conditions. Additionally, a new algorithm that computes globally optimal impulsive control input sequences for the described class of problems is developed. The algorithm combines the advantages of previous approaches based on primer vector theory and reachable set theory to mitigate sensitivity to numerical errors and minimize computation cost at every step. The algorithm is validated through implementation in challenging example problems based on the recently proposed Miniaturized Distributed Occulter/Telescope small satellite mission, which requires periodic formation reconfigurations in a perturbed, eccentric orbit with time-varying attitude constraints. Additionally, the suitability of the algorithm for real-time applications is demonstrated through a characterization of the run time on an embedded microprocessor for nanosatellites.

Keywords: linear optimal control, impulsive control, real-time trajectory planning

1. Introduction
The problem of minimizing the cost of a set of impulsive control inputs subject to the constraint that a linear time-variant dynamical system is driven to a specified state at a specified time has received a great deal of attention in literature. The distinguishing feature of impulsive control problems is that the magnitude of the control input is not constrained. However, a wide range of continuous control problems can be approximated as impulsive provided that the durations of the time intervals over which control input is applied are small. Because of this property, similar problems have been studied in a wide range of fields including engineering (Gaias, D’Amico & Ardaens, 2015; Chernick & D’Amico, 2018), epidemiology (Verriest, Delmotte & Egerstedt, 2005), and finance (Bouchard, Dang & Leblond, 2011). Indeed, this class of problem has been studied for over fifty years in the context of spacecraft rendezvous and formation flying (Lawden, 1963). The space community’s interest in these problems is motivated by the fact that spacecraft propellant is limited and cannot be replenished after launch. As a result, improving the efficiency of maneuver planning algorithms can significantly extend mission lifetimes. Additionally, the dynamics of the space environment are well-understood and can be accurately approximated by linear models (Sullivan, Grimberg & D’Amico, 2017).

Solution methodologies in literature for this problem can be divided into three broad categories: closed-form solutions, direct optimization, and indirect optimization methods. Closed-form solutions are highly desirable because they are robust, predictable, and computationally efficient. However, such solutions are inherently specific to the prescribed state representation, dynamics model, and cost functional. Indeed, such solutions have only been found to date for simple problems in spacecraft formation flying (D’Amico, 2010; Gaias, D’Amico & Ardaens, 2015; Chernick & D’Amico, 2018; Serra, Arzelier & Rondepierre, 2018). Direct optimization methods offer a greater degree of generality by formulating the optimal control problem as a nonlinear program with the times, magnitudes, and directions of the applied control inputs as variables (Betts, 1998). However, the minimum cost is generally a non-convex function of the times of the control actions (Sobiesiak & Dameret, 2015). As a result, such methods generally find only a local minimum and cannot guarantee convergence to a globally optimal solution. Some authors have sought to mitigate this issue by using genetic algorithms or multiple initial guesses to identify multiple candidate local minima (Kim & Spencer, 2002; Kim, Woo, Park & Choi, 2010).
2009), but these approaches still fail to guarantee convergence to a global minimum.

In light of these weaknesses, the majority of numerical approaches in literature are based on indirect optimization techniques that leverage properties of a primal/dual pair of optimization problems. The majority of these approaches are based on some form of the so-called “primer vector” in Lawden (1963), which is an alias for the part of the costate that governs the control input according to Pontryagin’s maximum principle. Using this method, the optimal control problem is cast as a two-point boundary value problem where an optimal solution must satisfy a set of analytical conditions on the evolution of the primer vector. While this approach has been studied continuously for over fifty years (Handelsman & Lion, 1968; Prussing, 1969, 1970; Jezewski, 1980; Roscoe, Westphal, Griesbach & Schaub, 2013), most studies in literature rely on an initial estimate of the number and times of control inputs. This estimate is refined until an analytical criteria is satisfied to add or remove a control input. An algorithm of this type was proposed in Roscoe et al. (2013) for spacecraft formation reconfigurations in perturbed, eccentric orbits. However, the algorithm is known to have a limited radius of convergence because it models the cost of a control input as the square of its 2-norm. As a result, the optimal cost varies with the number of allowed control inputs. Instead, the algorithm proposed in Arzelier et al. (2016) provides guaranteed convergence to a globally optimal solution using an iterative approach based on successive discretizations of the time domain. Specifically, this algorithm starts with a minimal set of candidate times for control inputs and adds a candidate time at each iteration until the optimality conditions are satisfied to within a user-specified tolerance. However, the algorithm is developed under two limiting assumptions: 1) the cost of a control input is its $p$-norm, and 2) the columns of the control input matrix are linearly independent. Also, no considerations are made regarding the sensitivity of the cost of feasible solutions to errors in the control input times in corner cases. A different approach to indirect optimization based on reachable set theory was proposed by Gilbert in 1971 (Gilbert & Harasty, 1971). This approach provides guaranteed convergence to a globally optimal sequence of impulsive control inputs for problems where the cost functional is the integral of a time-invariant norm-like function of the control input. This degree of generality enables modeling of effects of constraints on the control system (e.g. attitude constraints on a spacecraft). However, for some unknown reason this approach has not been adopted by the aerospace industry. A common limitation of all of these methods is that the cost of a specified control input is not allowed to vary over time. An algorithm that provides optimal solutions without this constraint could be applied to problems with complex, time-varying behaviors (e.g. attitude modes on a spacecraft). In addition to the obvious applications in trajectory planning for robots, spacecraft, and other systems, such an algorithm could provide optimal reference solutions that can be used to characterize the performance of simpler control laws.

To meet this need, this paper proposes a simple, robust, and efficient solution methodology for a broad class of optimal impulsive control problems for linear time-variant systems with only two additional assumptions: 1) the cost functional can be expressed as the integral of a time-varying norm-like function of the control input vector, and 2) no constraints are imposed on the state at intermediate times. Unlike many previous studies, this approach includes no domain-specific assumptions and can be applied to any system as long as the required matrices and cost function can be evaluated. The contributions of this paper to the state-of-the-art are threefold. First, necessary and sufficient optimality conditions are derived for the aforementioned class of optimal control problems. This derivation recovers all of the main findings of primer vector theory in Lawden (1963) for impulsive control input profiles (under the same additional assumptions) while providing a geometric interpretation of the meaning of the dual variable. Second, a method of quickly computing a lower bound on the minimum cost is proposed based on solutions to a relaxed problem. Third, a new three-step algorithm is proposed to compute globally optimal impulsive control input profiles. First, an initial set of candidate times for control inputs is computed from an a-priori estimate of the optimal dual variable. Second, the set of candidate times and dual variable are iteratively refined until the optimality conditions are satisfied to within a user-specified tolerance. Third, a globally optimal impulsive control input profile is computed. The geometry of the problem is exploited at every step to ensure robustness to corner cases and minimize computation cost. The algorithm is validated in three steps. First, the performance of the algorithm is demonstrated through implementation in a challenging example problem based on the proposed Miniaturized Distributed Occulter/Telescope (mDOT) small satellite mission (Koenig, D’Amico, Macintosh & Titus, 2015). Second, a Monte Carlo experiment is performed to demonstrate the robustness of the algorithm and characterize the number of required iterations for a wide range of problems. The sensitivity of the computation cost to errors in the initial estimate is characterized by solving each problem with three different initialization schemes. Third, the suitability of the algorithm for real-time applications is demonstrated through deployment on an embedded micro-processor for nanosatellites. Overall, this paper provides a simple and robust solution methodology for a challenging class of optimal impulsive control problems. The proposed algorithm can be used in a wide range of real-time applications and may enable new capabilities such as onboard optimal trajectory planning for spacecraft formations.

Following this introduction, the optimal control problem is formally defined in Section 2. Next, the optimization
If no constraints are imposed on the magnitude of the control input vector and the desired pseudostate is reachable, then there must exist an optimal control input profile consisting of a set of \( k \) impulses that can be expressed as

\[
u(t) = \sum_{j=1}^{k} u_j \delta(t - t_j) \quad (7)
\]

where \( \delta \) denotes the Dirac delta function. It should be noted that that the optimal control input profile is not necessarily unique. Under the assumption that the control input profile takes the form in (7), the optimal control problem can be reformulated as

\[
\begin{align*}
\text{minimize:} \quad & c = \sum_{j=1}^{k} f(u_j, t_j) \\
\text{subject to:} \quad & w = \sum_{j=1}^{k} \Gamma(t_j)u_j, \quad t_j \in T
\end{align*}
\]

In this paper it is assumed that the cost function \( f(u, t) \) is a time-varying norm-like function that has the following two properties: 1) all sublevel sets of \( f(u, t) \) at a specified time are convex and compact, and 2) \( f(\alpha u, t) = \alpha f(u, t) \) for any \( \alpha \geq 0 \). The first property ensures that the resulting optimization problem is convex. The second property means that the cost of a control input applied at a specified time scales linearly with its magnitude. This property ensures that the minimum cost can be achieved with a finite number of control inputs, as will be demonstrated in the following section. The most commonly considered cost function in literature is the \( p \)-norm, which is defined as

\[
f(u, t) = ||u||_p = \left\{ \begin{array}{ll}
\left( \sum_{j=1}^{m} |u_j|^p \right)^{1/p}, & 1 \leq p < \infty \\
\max_{j\in[1,\ldots,m]} |u_j|, & p = \infty
\end{array} \right.
\]

This cost function has been the focus of the majority of studies using primer vector theory (Lawden, 1963; Neustadt, 1964; Carter & Brient, 1995). However, all \( p \)-norms are highly symmetric. There are many noteworthy problems for which the actual control cost exhibits complex behavior due to constraints imposed on the system. For example, a spacecraft may have multiple attitude modes that affect the efficiency of executed maneuvers. Some examples of these cost functions with corresponding attitude constraints are included in Table II. Additionally, these constraints may change over time, resulting in a time-varying cost function. The algorithm proposed in this paper can handle any such cost function as long as it meets the two aforementioned requirements.

## 3. Formulation of the Optimization Problem

The optimal control problem defined in (8) can be reformulated as a globally convex problem using the geometric...
It is evident from (12) that \( S(c, t) \) is simply the convex hull of \( S(c, t) \).

Using the provided set definitions, an equivalent optimal control problem to that in (8) can be posed as follows: minimize \( c \) subject to the constraint that \( w \) is reachable. A mathematical formulation of this problem is given by

\[
\text{minimize: } c \quad \text{subject to: } w \in S^*(c, T) \tag{15}
\]

Because \( S^*(c, T) \) is convex, it must be possible to express it as the intersection of a set of half-spaces (B. Boyd & Vandenberghe, 2004). Thus, an equivalent form of (15) is given by

\[
\text{minimize: } c \quad \text{subject to: } \max_{z \in S^*(c, T)} \hat{\eta}^T z \geq \hat{\eta}^T w \quad \forall \hat{\eta} \in \mathbb{R}^n, ||\hat{\eta}||_2 = 1
\tag{16}
\]

It is noteworthy that this problem formulation does not explicitly include the control inputs. Instead, they are included implicitly in the requirement that \( z \) is in \( S^*(c, T) \). Also, this problem formulation has a simple geometric interpretation. Suppose that \( \hat{\eta} \) denotes the unit normal of a hyperplane. The left side of the constraint in (16) is the distance this hyperplane must be from the origin so that it is tangent to \( S^*(c, T) \). Thus, the constraint requires that \( w \) and \( S^*(c, T) \) are on the same side of any hyperplane that is tangent to \( S^*(c, T) \). Additionally, it is obvious that for an optimal \( c \), there must exist at least one \( \hat{\eta} \) such that \( \max_{z \in S^*(c, T)} \hat{\eta}^T z = \hat{\eta}^T w \). Because \( w \) must be in \( S^*(c, T) \), it is evident that \( c \) is the minimum cost to reach \( w \) if and only if \( w \) is in the boundary of \( S^*(c, T) \).

However, the optimization problem in (16) is computationally intractable because the constraint equation must be satisfied for all unit vectors \( \hat{\eta} \). This issue can be remedied using simple geometry. If \( w \) is not in the interior of \( S^*(c, T) \), there must exist at least one supporting hyperplane to \( S^*(c, T) \) that contains \( w \) because \( S^*(c, T) \) is convex (B. Boyd & Vandenberghe, 2004). Using this property, a dual of the optimization problem in (16) can be posed as follows: maximize \( c \) subject to the constraint that there exists a supporting hyperplane to \( S^*(c, T) \) that contains \( w \). This dual problem can be formulated as

\[
\text{maximize: } c \quad \text{subject to: } \max_{z \in S^*(c, T)} \chi^T z \leq \chi^T w \quad \chi \in \mathbb{R}^n
\tag{17}
\]

In this formulation, \( \chi \) is a vector that defines the normal of a supporting hyperplane to \( S^*(c, T) \). It should be noted that \( \chi \) appears on both sides of the constraint equation. Consequently, if the constraint is satisfied for a given \( \chi \), then it must also be satisfied for any positive scalar multiple of \( \chi \). This means that only the direction of \( \chi \) is important. Also, as opposed to the formulation in (16), the constraint equation in this problem only needs to be satisfied for a single \( \chi \). Additionally, because the primal and dual problems are formulated as linear programs, they must have the same optimal objective value.

### Table 1: Example cost functions and associated constraints.

| \( f(u, t) \) | Associated constraints |
|----------------|----------------------|
| \( ||u||_2 \)  | Spacecraft can align a single thruster in any direction |
| \( u_1 + \sqrt{u_2^2 + u_3^2} \) | Spacecraft with two perpendicular thrusters, one with a fixed direction |
| \( \max(Cu) \)  | Spacecraft with fixed attitude and multiple thrusters in an asymmetric configuration |
Thus, the minimum cost can be computed by solving the more tractable dual problem. The dual problem can be simplified by expanding the constraint function. Because \( S^*(c, T) \) is the convex hull of \( S(c, T) \), the left side of the constraint in (17) can be reformulated as
\[
\max_{x \in S^*(c, T)} \lambda^T z = \max_{y \in S(c, T)} \lambda^T y.
\]

Using the set definitions in (10-12) and the linearity of the constraint function, 
\( \lambda^T z \) can be computed by solving the more tractable dual problem given by
\[
\max_{y \in S(c, T)} \lambda^T y = \max_{t \in T} \left( \max_{y \in S(c, t)} \lambda^T y \right) = \max_{t \in T} \left( \max_{u \in U(1, t)} \lambda^T \Gamma(t) u \right) = c \left( \max_{t \in T} \left( \max_{u \in U(1, t)} \lambda^T \Gamma(t) u \right) \right)
\]

Using the substitutions in (18-19), the dual problem in (17) can be reformulated as an unconstrained optimization problem given by
\[
\text{maximize: } \frac{\lambda^T w}{\max_{t \in T} \left( \max_{u \in U(1, t)} \lambda^T \Gamma(t) u \right)}
\]

This form of the dual problem is used to rapidly compute lower bounds on the minimum cost as described in Section 5. Also, it is obvious that \( \max_{u \in U(1, t)} \lambda^T \Gamma(t) u \) must be nonnegative and finite at all times because it is always possible to set \( u \) as zero and all sublevel sets of \( f(u, t) \) are convex and compact. It follows that the minimum cost to reach any nonzero \( w \) must be positive, which means that \( \lambda^T w > 0 \) for an optimal \( \lambda \). Additionally, it is evident that the minimum cost to reach any \( w \) is linearly proportional to its magnitude. Using simple algebraic manipulation, the dual problem can also be formulated as
\[
\text{maximize: } \lambda^T w \\
\text{subject to: } \max_{t \in T} \left( \max_{u \in U(1, t)} \lambda^T \Gamma(t) u \right) \leq 1
\]

which closely resembles the semi-infinite convex problem in Neustadt (1964). This form of the dual problem enables development of the algorithm described in Section 6.

4. Optimality Conditions

The geometric relationships between the reachable sets and the supporting hyperplane can be used to derive necessary and sufficient optimality conditions for impulsive control input profiles as demonstrated in the following. First, consider the optimality conditions for the dual variable \( \lambda \). For simplicity, it is assumed in the following that \( \lambda^T w \geq 0 \), which must be true for an optimal \( \lambda \) as demonstrated in the previous section. Let \( L(w, \lambda) \) denote the hyperplane that contains \( w \) and is perpendicular to \( \lambda \). Also, suppose that \( c \) is always selected such that \( L(w, \lambda) \) is a supporting hyperplane to \( S^*(c, T) \). It follows that every point in \( L(w, \lambda) \cap S^*(c, T) \) is in the boundary of \( S^*(c, T) \). Also, recall that \( c \) is the minimum cost to reach a given \( w \) if and only if \( w \) is in the boundary of \( S^*(c, T) \).

An immediate consequence of these properties is that \( w \) must be in \( L(w, \lambda) \cap S^*(c, T) \) for an optimal choice of \( \lambda \). Indeed, if \( w \) is not in \( L(w, \lambda) \cap S^*(c, T) \), then it cannot be in \( S^*(c, T) \), which means that it cannot be reached at a cost of \( c \). Thus, a necessary and sufficient optimality condition for \( \lambda \) is that it is an outward normal direction to \( S^*(c, T) \) at \( w \). It should be noted that the optimal \( \lambda \) may not be unique if \( w \) lies on an edge or vertex of \( S^*(c, T) \). This optimality condition is illustrated in Figure 1 (left) for a three-dimensional example system. In this figure, \( w \) is shown in black, \( S(c, T) \) is shown in blue, \( S^*(c, T) \) is shown in translucent red, the optimal \( \lambda \) is shown as a green arrow, and \( L(w, \lambda) \) is shown as a gray plane. It is evident that \( w \) is in the boundary of \( S^*(c, T) \), which means that \( c \) is the minimum cost to reach \( w \). Also, \( \lambda \) is an outward normal to \( S^*(c, T) \) at \( w \) and the corresponding hyperplane is tangent to \( S^*(c, T) \).

This geometric relationship can also be used to determine an upper bound on the number of required control inputs. Because \( S^*(c, T) \) is the convex hull of \( S(c, T) \), it must be possible to express any \( w \) in \( L(w, \lambda) \cap S^*(c, T) \) as a convex combination of points in \( L(w, \lambda) \cap S(c, T) \).

Figure 1: Illustration of the optimality conditions for dual variable (left) and control inputs (right).
property is illustrated in Figure 1 (right). It is evident that $L(w, \lambda)$ is tangent to $S(c, T)$ at three points (shown in purple) and that $w$ is in the convex hull of these points. In this example, $w$ can be expressed as a unique combination of these intersection points, meaning that the optimal control input profile is unique (if $f(u, t)$ and $\Gamma(t)$ are not periodic). However there may exist scenarios with many more points in $S(c, T) \cap L(w, \lambda)$. In these cases, it must be possible to express $w$ as a convex combination of a subset of $n$ or fewer of these points.

Next, it is necessary to derive necessary and sufficient optimality conditions on the control input profile. It is evident from the preceding discussion and Figure 1 that a control input profile that reaches $L(w, \lambda)$ at minimum cost for an optimal $\lambda$. This with in mind, it is instructive to consider the relaxed problem of minimizing the cost to reach any point in $L(w, \lambda)$ for a specified $\lambda$. This relaxed problem can be expressed as

\begin{equation}
\text{minimize: } \sum_{j=1}^{k} f(u_j, t_j) \\
\text{subject to: } \lambda^T w = \sum_{j=1}^{k} \lambda^T \Gamma(t_j) u_j \quad t_j \in T
\end{equation}

It is assumed in the following that $L(w, \lambda)$ is reachable. Also, the function $g(u, \lambda, t)$ is defined as

\begin{equation}
g(u, \lambda, t) = \lambda^T \Gamma(t) u
\end{equation}

to simplify notation. Because the relaxed problem in (22) has a single constraint, it is possible to reach $L(w, \lambda)$ using a single control input applied at any time when $g(u, \lambda, t) > 0$ for some admissible $u$. The cost of this control input is minimized if it is selected to maximize the ratio $g(u, \lambda, t)/f(u, t)$, which is equivalent to $\max_{u \in U(1,t)} g(u, \lambda, t)$. Thus, the minimum cost to reach $L(w, \lambda)$ using a single control input executed at time $t$, denoted $c_\lambda(t)$, is given by

\begin{equation}
c_\lambda(t) = \frac{\lambda^T w}{\max_{u \in U(1,t)} g(u, \lambda, t)}
\end{equation}

Additionally, the global minimum cost is achieved by only applying control input when $\max_{u \in U(1,t)} g(u, \lambda, t)$ takes its maximum value over the domain $T$. The resulting minimum cost, denoted $c_\lambda$, is given by

\begin{equation}
c_\lambda = \max_{t \in T} \left( \frac{\lambda^T w}{\max_{u \in U(1,t)} g(u, \lambda, t)} \right)
\end{equation}

Next, suppose that it is known that $\max_{u \in U(1,t)} g(u, \lambda, t)$ takes on its maximum value at a set of times $T_{opt}$. It is possible to reach $L(\lambda, w)$ at minimum cost by applying a single impulsive control input $u_{opt}$ at any time in $T_{opt}$ of the form

\begin{equation}
u_{opt} = c_\lambda \arg \max_{u \in U(1,t)} g(u, \lambda, t)
\end{equation}

Any convex combination of these control inputs will also reach $L(\lambda, w)$ at minimum cost. It follows that any optimal control input profile $u_{opt}(t)$ must be of the form

\begin{equation}
u_{opt}(t) = \begin{cases}
\alpha(t) \arg \max_{u \in U(1,t)} g(u, \lambda, t), & t \in T_{opt} \\
0, & \text{otherwise}
\end{cases},
\end{equation}

Using this result, necessary and sufficient optimality conditions for a control input profile are defined as follows for any $\lambda$ corresponding to an optimal solution to any of the equivalent optimization problems in (17), (20), and (21): 1) control input is only applied at times when $\max_{u \in U(1,t)} g(u, \lambda, t)$ takes on its maximum value over the domain $T$, and 2) any applied control input must be of the form $\alpha(t) \arg \max_{u \in U(1,t)} g(u, \lambda, t)$ for some $\alpha(t) \geq 0$. It should be noted that the direction of the optimal control input vector is not necessarily unique for a specified time. For example, if sublevel sets of $f(u, t)$ at a specified time are polyhedra, then $L(\lambda, w)$ may be tangent to $S(c, t)$ at multiple points. To use the algorithm described in Section 6, it is sufficient to include only the vertices of $L(\lambda, w) \cap S(c, t)$. If this set has an infinite number of vertices (e.g. if the boundary of $L(\lambda, w) \cap S(c, t)$ is a circle), then it will be necessary to approximate the boundary using a finite number of points. However, none of the cost functions considered in this paper require this approximation.

The meaning of these optimality conditions can be understood by considering Figure 1 (right). In this example there are three times in $T_{opt}$ (assuming that $f(u, t)$ and $\Gamma(t)$ are not periodic). If a single impulsive control input is applied, then the control input profile will drive the pseudostate to one of the points shown in purple. Instead, by applying a combination of impulses at the three optimal times with a total cost of $c_\lambda$, it is possible to reach any point in the triangle spanned by the points in $L(\lambda, w) \cap S(c, T)$. From this behavior, it is evident that the times at which optimal control inputs can be applied and admissible directions can be determined directly from an optimal $\lambda$ and $f(u, t)$, but the magnitudes of the control inputs must still be computed to reach a specified target pseudostate.

It is worthwhile to compare these optimality conditions with those developed in Lawden (1963) for impulsive control profiles. Lawden’s necessary and sufficient conditions are formulated with respect to a so-called “primer vector” which is an alias for the part of the costate that governs the control input according to Pontryagin’s maximum principle. Using the notation adopted in this paper, the primer vector is equivalent to $\Gamma^T(t) \lambda$. However, Lawden
addressed a restricted problem where the cost of a control input is equal to its Euclidean norm and the control input matrix is assumed constant. Under this assumption, the Cauchy-Schwarz inequality provides

\[
\max_{u \in U(1,t)} \lambda^T \Gamma(t) u = \|\Gamma^T(t) \lambda\|_2,
\]

\[
\arg \max_{u \in U(1,t)} \lambda^T \Gamma(t) u = \frac{\Gamma^T(t) \lambda}{\|\Gamma^T(t) \lambda\|_2}
\] \hspace{1cm} (28)

Lawden’s necessary and sufficient conditions can be summarized as three constraints on the primal vector: 1) the primal vector and its first derivative are continuous everywhere, 2) the primal vector must have a constant magnitude \( P \) whenever control input is applied, and 3) the magnitude of the primal vector cannot exceed \( P \) at any time at which control input is allowed. Because the primal vector evolves according to \( \Gamma(t) \), it is evident that the continuity of the primal vector and its first derivative are due to the assumptions that the cost is the 2-norm of the control input vector and the control input matrix is constant. Also, it is evident from the relationships in (28) that Lawden’s second and third conditions are equivalent to the conditions provided in this paper under the assumption that the cost function is the 2-norm of the control input vector. The necessary and sufficient conditions posed by other authors \cite{Neustadt1964, Gilbert1971, Carter1995} can be recovered in the same manner by applying the appropriate assumptions. However, the necessary and sufficient conditions for optimality provided in this paper are applicable to a more general class of optimal control problem with a time-varying cost function.

5. Rapid Computation of Lower Bounds

Some fields may benefit from an algorithm that rapidly computes a lower bound on the minimum cost to reach a specified \( w \) instead of solving the complete optimal control problem. Such a lower bound could be used to quickly sub-optimize of proposed control laws. This tool would be especially valuable in applications with strict limitations on available computing power. Indeed, several publications have addressed the problem of finding lower bounds on the cost of spacecraft formation reconfigurations based on analytical properties of \( \Phi(t) \) and \( B(t) \) \cite{DAmico2010, Gaia2013, Chernick2018}. However, the procedure presented in the following is more general because it can be applied to any linear system, accommodates a wider range of cost functions, and can provide an arbitrarily accurate approximation of the reachable lower bound.

A useful property of primal/dual pairs of optimization problems is that any feasible value of the objective of a dual problem is a lower bound on the optimal objective of the primal problem. Thus, a lower bound can be computed by simply evaluating the objective of the unconstrained dual problem in (20) for any \( \lambda \). However, the lower bound is only of value in practice if it is reasonably close to the minimum cost to reach \( w \). With this in mind, the gap between the lower bound and the minimum cost can be reduced by considering multiple choices of \( \lambda \). If \( \Lambda \) denotes a set of user-specified \( \lambda \), then an improved lower bound \( c_A \) is given by

\[
c_A = \max_{\lambda \in \Lambda} \frac{\lambda^T w}{\max_{t \in T} \left( \max_{u \in U(1,t)} \lambda^T \Gamma(t) u \right)}
\]

This lower bound can be evaluated by performing a finite number of global searches over the domain \( T \) provided that \( \max_{u \in U(1)} g(u, \lambda, t) \) can be evaluated. This formula also has a useful geometric interpretation that is a natural extension of the supporting hyperplane interpretation of the dual problem. Specifically, the vectors in \( \Lambda \) describe the outward face normals of a polyhedron that circumscribes \( S^*(c_A, T) \). It is evident that including more elements in \( \Lambda \) generally reduces the gap between the lower bound and the minimum cost to reach \( w \), but increases the computational cost of evaluating \( c_A \). It follows that the practical value of this approach relies on the ability to produce a reasonable approximation of the reachable set using only a small number of elements of \( \Lambda \). This goal is accomplished by properly selecting the elements of \( \Lambda \) to maximize the lower bound. From the structure of (29), each \( \lambda \) should be selected to maximize the numerator, minimize the denominator, or some combination of the two. It should be noted that because both the numerator and denominator both include \( \lambda \), it is only possible to change the lower bound by changing the direction of \( \lambda \). The numerator can be maximized by simply selecting a \( \lambda \) that is parallel to \( w \). The denominator can be minimized by incorporating domain-specific knowledge on the behavior of \( \Gamma(t) \). For example, if it is known that the elements of one row of \( \Gamma(t) \) are much larger than the elements of other rows, it is reasonable to expect the corresponding component of \( \lambda \) to be small to minimize the denominator of (29). However, the behavior of \( \Gamma(t) \) depends on the chosen state definition and dynamics model for real systems. As such, a proper choice of the state representation can improve the accuracy of the lower bound computed using this approach.

6. An Efficient and Robust Control Algorithm

Leveraging the geometric properties of the dual problem demonstrated in the previous sections, an efficient and robust algorithm to compute a globally optimal sequence of impulsive control actions for any linear time-variant system is proposed in the following. This algorithm includes three steps: 1) generation of an initial set of candidate control input times, 2) iterative refinement of the set of candidate times and computation of an optimal dual variable, and 3) extraction of optimal control inputs.
Initialization of Control Input Times

The first step in the algorithm is the generation of a set of candidate control input times $T_j$. The only requirement imposed on this step is that $w$ is reachable using a set of admissible control inputs applied at times in $T_j$. However, to minimize the number of required iterations in the refinement procedure, it is desirable to select these times according to a criteria that produces a reasonable approximation of an optimal solution. This can be accomplished by using an a-priori estimate of $\lambda$, denoted $\lambda_{est}$. From the behavior of the lower bound described in Section 3, a reasonable choice of $\lambda_{est}$ is a vector parallel to $w$ or a transformation of such a vector according to a heuristic model based on the known behavior of $\Gamma(t)$. An initial set of candidate times for control inputs can be obtained as follows. First, a set of times $T_d$ is computed from a uniform discretization of $T$. Next, $\max_{u \in U(1,t)} g(u, \lambda_{est}, t)$ is computed for each time in $T_d$. The initial set of control input times is chosen as the $k$ times in $T_d$ at which $\max_{u \in U(1,t)} g(u, \lambda_{est}, t)$ is largest. This initialization approach is summarized in the following pseudocode.

Algorithm 1: Initialization

Inputs: $T_d, \lambda_{est}, \Gamma(t), k$  
Outputs: $T_j$

compute $\max_{u \in U(1,t)} g(u, \lambda_{est}, t)$ for each $t \in T_d$
$T_{sort} \leftarrow t \in T_d$ sorted in descending order by $\max_{u \in U(1,t)} g(u, \lambda_{est}, t)$
$T_j \leftarrow$ first $k$ elements of $T_{sort}$
return $T_j$

A notional example of this initialization procedure is shown in Figure 2 for a two-dimensional system. In this example, $T_d$ includes four times (indicated by vertical lines in the left plot), and the algorithm must select the two best times. The two selected times (indicated by circles in the left plot) are those at which $\max_{u \in U(1,t)} g(u, \lambda_{est}, t)$ is largest. The rejected candidates are indicated by x markers in the left plot. The reachable sets $S(c,t)$ for each of these times are shown in the right plot. The solid lines indicated the selected times and the dashed lines indicate the rejected times. It is evident that the reachable sets at selected times include points with the largest possible dot product with the vector $\lambda_{est}$.

Iterative Refinement of Dual Variable and Times

The initial dual variable and set of control input times must be refined until two objectives are met before an optimal control input profile can be extracted. First, an optimal dual variable $\lambda_{opt}$ and cost $c_{opt}$ must be computed such that $\lambda_{opt}$ is an outward normal direction to $S(c_{opt}, T)$ at $w$. Second, an optimal set of optimal control input times $T_{opt}$ must be found such that $w$ is in the boundary of both $S(c_{opt}, T)$ and $S'(c_{opt}, T_{opt})$. To meet this need, the refinement procedure described in the following provides monotonic convergence to a solution with a cost within a user-specified threshold of the global optimum. This procedure is similar to the algorithm in Arzelier et al. (2016), but includes modifications to minimize the number of constraints that must be enforced in the required optimization problems and reduce the number of required iterations. Let $T_j$ denote the current iterate of the set of candidate control times, $\lambda_j$ denote the current iterate of the dual variable. First, the algorithm solves the constrained formulation of the dual problem in (21) with the modification that control input is only allowed at times in $T_j$. The optimal objective of this problem, denoted $c_j$, is the cost of a feasible solution, which is an upper bound on the minimum cost to reach $w$. Using the $\lambda_j$ corresponding to this solution, all $t \in T_j$ that satisfy

$$\max_{u \in U(1,t)} g(u, \lambda_j, t) < 1 - \epsilon_{remove}$$

(30)

for user-specified tolerance $\epsilon_{remove}$ are removed from $T_j$ to reduce the computational effort of subsequent iterations. Removing these times has no impact on the cost because optimal control inputs cannot be applied at these times. Next, $\max_{u \in U(1,t)} g(u, \lambda_j, t)$ is evaluated for all times in $T$. All times of local maxima of $\max_{u \in U(1,t)} g(u, \lambda_j, t)$ that are greater than one are added to $T_j$ to reduce the number of required iterations. If $\max_{u \in U(1,t)} g(u, \lambda_j, t) \leq 1 + \epsilon_{cost}$ at all times in $T$ for a user-specified threshold $\epsilon_{cost}$, then the algorithm terminates. This ensures that $c_j$ is within a factor of $\epsilon_{cost}$ of the lower bound computed using (25). The described iteration procedure is summarized in the following pseudocode.

Algorithm 2: Iterative Refinement

Inputs: $T_j, T, w, \Gamma(t), \epsilon_{cost},$ and $\epsilon_{remove}$
Outputs: $T_{opt}, \lambda_{opt},$ and $c_{opt}$

do
$c_j, \lambda_j \leftarrow$ solution of problem:
maximize: $c = \lambda^T w$
subject to: $\max_{t \in T_j} (\max_{u \in U(1,t)} g(u, \lambda, t)) \leq 1$

loop $t \in T_j$
if $\max_{u \in U(1,t)} g(u, \lambda_j, t) < 1 - \epsilon_{remove}$
remove $t$ from $T_j$
loop local maxima of $\max_{u \in U(1,t)} g(u, \lambda_j, t)$ in $T$
if \( \max_{u \in U(1,t)} g(u, \lambda_j, t) > 1 \)
add \( t \) to \( T_j \)
while \( \max_{l \in T} \max_{u \in U(1,t)} g(u, \lambda_j, t) > 1 + \epsilon_{\text{cost}} \)
\( T_{\text{opt}} \leftarrow T_j \)
\( \lambda_{\text{opt}} \leftarrow \lambda_j \)
\( c_{\text{opt}} \leftarrow c_j \)
return \( T_{\text{opt}}, \lambda_{\text{opt}}, c_{\text{opt}} \)

This refinement procedure ensures that \( c_j \) decreases monotonically towards the global minimum cost. To understand this property, consider the notional iteration illustrated in Figure 3. In this example, the set of candidate times used to compute \( \lambda_j \) is indicated by solid vertical lines. It is evident that the \( \max_{u \in U(1,t)} g(u, \lambda_j, t) \leq 1 \) at all of these times. However, \( \max_{u \in U(1,t)} g(u, \lambda_j, t) \leq 1 - \epsilon_{\text{remove}} \) for two of these times (indicated by \( x \)). These times are removed from \( T_j \). Next, the times of local maxima of \( \max_{u \in U(1,t)} g(u, \lambda_j, t) \) that are greater than one (indicated by triangles) are added to \( T_j \). Using this new set of times \( T_{j+1} \), the dual variable \( \lambda_{j+1} \) and cost \( c_{j+1} \) are recomputed. The evolution of \( \max_{u \in U(1,t)} g(u, \lambda_{j+1}, t) \) for is shown as a dashed line. This new solution must satisfy \( c_{j+1} \leq c_j \) because \( \lambda_j \) is not a feasible solution to the dual problem in [21] if control inputs are allowed at times in \( T_{j+1} \). While no rigorous guarantee is provided for the speed of convergence, the results in Section 7 demonstrate that a wide range of problems can be solved in less than ten iterations.

Figure 3: Illustration of criteria for updating \( T_j \) including removed times (\( x \)) and added times (triangle).

*Extraction of Optimal Control Inputs*

Once a set of optimal control input times \( T_{\text{opt}} \) and dual variable \( \lambda_{\text{opt}} \) are obtained, it is necessary to compute a set of optimal control inputs. To mitigate the known sensitivity of the cost of a control input sequence to errors in the execution times in corner cases, the extraction algorithm computes the point in the convex cone of candidate optimal control inputs that is closest to the desired pseudostate. Provided that \( \lambda_{\text{opt}} \) is properly computed (i.e. the solver used in the iterative refinement algorithm converged), the residual error will be negligible for practical applications. Additionally, the objective is formulated as the quadratic product of the error vector and a user-specified positive definite weight matrix \( Q \) to ensure well-behaved solutions. This optimal control input extraction algorithm is described in the following pseudocode.

**Algorithm 3: Control Input Extraction**

**Inputs:** \( T_{\text{opt}}, \lambda_{\text{opt}}, \Gamma(t), w, Q \)

**Outputs:** \( u_{\text{opt}}(t) \forall t \in T_{\text{opt}} \)

**loop** \( t_j \in T_{\text{opt}} \)

\[ u_{\text{opt}}(t_j) \leftarrow \arg \max_{u \in U(1,t_j)} \lambda_{\text{opt}}^T \Gamma(t_j) u \]

\[ y_j \leftarrow \Gamma(t_j) u_{\text{opt}}(t_j) \]

\( \alpha \leftarrow \) solution to optimization problem:

minimize: \( w_{\text{err}}^T Q w_{\text{err}} \)

subject to: \( w_{\text{err}} = w - \sum \alpha_j y_j \), \( \alpha_j \geq 0 \)

**loop** \( t_j \in T_{\text{opt}} \)

\[ u_{\text{opt}}(t_j) \leftarrow \alpha_j u_{\text{opt}}(t_j) \]

**return** \( u_{\text{opt}}(t) \forall t \in T_{\text{opt}} \)

A notional example of the optimal control input extraction algorithm is shown in Figure 4 for a two-dimensional system. In this example, there are two candidate times for optimal control inputs. First, the optimal maneuver directions \( u_{\text{opt}} \) are selected such that \( L(w, \lambda_{\text{opt}}) \) can be reached by a single control input of cost \( c_{\text{opt}} \) at either of these times as shown in the left plot. Next, a positive linear combination of these control inputs is computed that reaches the specified \( w \) at minimum cost as shown in the right plot.

Figure 4: Illustration of example optimal control input extraction for two-dimensional example including computation of optimal control input directions (left) and computation of scaling factors (right).

*Summary*

Using the described algorithm, a globally optimal impulsive control input sequence can be computed in three steps. First, an initial set of candidate times for control inputs is selected. The only requirement on the initialization is that the target pseudostate must be reachable using control inputs at the specified times. However, an a-priori estimate of the dual variable can be used to generate a better set of candidate times, reducing computation cost of subsequent steps. Second, the set of candidate times and dual variable
are refined using an iterative process until the optimality criteria are satisfied to within a user-specified tolerance. Each iteration requires a single globally convex optimization problem to be solved. The optimization problem has a linear objective, but the complexity of the constraints depends on the definition of the cost function. The results in the following section demonstrate that a wide range of problems can be solved in less than ten iterations. Third, an optimal control input sequence is obtained from the final set of control input times and optimal dual variable by solving a simple quadratic program.

7. Numerical Validation

The proposed algorithm is validated through application to a number of challenging spacecraft formation reconfiguration problems based on the Miniaturized Distributed Occulter/Telescope (mDOT) small satellite mission recently proposed in [Koenig et al. (2015)]. The primary objective of this mission is to increase confidence in the optical and formation flying technologies needed for proposed full-scale exoplanet imaging missions. This mission includes a micro-satellite occulter and a CubeSat telescope spacecraft deployed in a highly eccentric earth orbit. At the apogee of each orbit, the formation achieves and maintains precise alignment with a target star so that the telescope can directly image objects in the vicinity of the star from within the shadow of the occulter. To ensure that the formation is aligned with the target star at the apogee of each orbit, periodic formation reconfigurations will be required. These reconfigurations are challenging for three reasons: 1) the formation is in a perturbed, eccentric orbit, 2) the reconfigurations must be accomplished in less than one orbit, and 3) the spacecraft are subject to time-varying attitude constraints to facilitate communication with ground stations. Also, the formation has a large nominal separation (500-1000 km) in the cross-track direction. Linear dynamical models in literature based on the relative position and velocity of the spacecraft cannot be used for this mission due to this large separation. However, linear models based on mean relative orbital elements (ROE) can accommodate such a large formation in a $J_2$-perturbed orbit provided that the separation is established primarily through changes in the right ascension of the ascending node (RAAN), argument of perigee, or mean anomaly as shown in [Koenig, Guffanti & D’Amico (2017)]. With this in mind, let $a$, $c$, $i$, $\Omega$, $\omega$, and $M$ denote the Keplerian orbit elements of a spacecraft. Specifically, $a$ is the semi-major axis, $c$ is the eccentricity, $i$ is the inclination, $\Omega$ is the RAAN, $\omega$ is the argument of perigee, and $M$ is the mean anomaly. Next, let the orbit element vector $\mathbf{e}$ be defined as

$$\mathbf{e} = [a \ c \ i \ \Omega \ \omega \ M]^T \quad (31)$$

To use the linear dynamics model provided in [Appendix A] it is necessary to distinguish between an osculating orbit, denoted $\mathbf{e}'$, and a mean orbit, denoted $\mathbf{e}$. The osculating orbit is the set of orbit elements computed from the instantaneous position and velocity of a spacecraft. However, these elements are subject to short period, long period, and secular variations due to the effects of perturbations. To simplify the dynamics, it is helpful to adopt a set of mean orbit elements that removes the short and long period oscillations due to $J_2$. The mean and osculating orbits are related by

$$\mathbf{e} = f_{\text{mean}}(\mathbf{e}') \quad (32)$$

where the function $f_{\text{mean}}$ is the analytical osculating to mean transformation for $J_2$-perturbed orbits described in [Brouwer (1959)]. To simplify computation, this paper uses the first-order truncation of this transformation in [Schaub & Junkins (2003)]. Use of this transformation allows the problem to be solved using the mean orbits of both spacecraft.

Next, it is necessary to define the relative state of the formation. The mean ROE state employed in these numerical simulations is defined with respect to the mean orbits of the chief, denoted by subscript $c$, and the deputy, denoted by subscript $d$, by

$$\mathbf{x} = h(\mathbf{e}_c, \mathbf{e}_d) = \begin{pmatrix} \delta a \\ \delta \lambda \\ \delta c_x \\ \delta c_y \\ \delta i_x \\ \delta i_y \end{pmatrix} = \begin{pmatrix} \Delta a / a_c \\ \Delta M + \eta_c (\Delta \omega + \Delta \Omega \cos(i_c)) \\ \epsilon_d \cos(\omega_d) - \epsilon_c \cos(\omega_c) \\ \epsilon_d \sin(\omega_d) - \epsilon_c \sin(\omega_c) \\ \Delta i \\ \Delta \Omega \sin(i_c) \end{pmatrix} \quad (33)$$

where $\eta = \sqrt{1 - e^2}$ and the operator $\Delta$ denotes the difference between the orbit elements of the deputy and chief (e.g. $\Delta a = a_d - a_c$). In this paper it is assumed that the telescope spacecraft is the chief and the occulter spacecraft is the deputy. It is also assumed that all maneuvers are performed by the occulter spacecraft. The linear dynamics model for mean ROE used to solve the following reconfiguration problems is provided in [Appendix A]. The state transition matrix is the same as the $J_2$-perturbed STM developed in [Koenig, Guffanti & D’Amico (2017)] for quasi-nonsingular ROE except that the second row is modified to accommodate the changed definition of $\delta \lambda$. The control input matrix is the same as that used in [Chernick & D’Amico (2015)]. The columns of the control matrix correspond to thrusts applied to the deputy spacecraft in the radial (R), along-track (T), and cross-track (N) directions, respectively. The R direction is aligned with the position vector of the spacecraft, the N direction is aligned with the angular momentum vector of the orbit, and the T direction completes the right-handed triad.

Next, it is necessary to define the parameters of the reconfiguration problem, the cost function, and algorithm parameters. The initial orbits of the spacecraft are selected be representative of a mission to image the vicinity of Beta Pictoris. The osculating and mean orbits for both spacecraft at the start of the reconfiguration are provided
in Table 2. All angular orbit elements (i, Ω, ω, and M) are expressed in radians.

| a (km) | e  | i   | Ω   | ω   | M    |
|--------|----|-----|-----|-----|------|
| 25001  | 0.700 | 0.680 | 6.251 | 6.261 | 3.409 |
| 24998  | 0.700 | 0.679 | 6.223 | 6.283 | 3.408 |
| 25003  | 0.700 | 0.680 | 6.251 | 6.261 | 3.409 |
| 25000  | 0.700 | 0.679 | 6.223 | 6.283 | 3.408 |

These orbits have a period of 11 hours. If the formation is required to image the target for one hour per orbit, then the formation reconfiguration must occur in ten hours. Thus, t_i is selected as 0 and t_f is selected as 36000 seconds. The control domain T is selected as a uniform discretization of the interval [t_i, t_f] with ten second intervals for a total of 3601 candidate times. Also, the occulter spacecraft will need to communicate with a ground station every orbit to downlink data from each observation. To accommodate this constraint, it is assumed that the occulter must maintain a fixed attitude in the RTN frame in the interval [4, 6] hours to facilitate communications with ground stations and no attitude constraints are enforced outside of this interval. It is noted that this interval is significantly longer than normal ground contacts. However, this choice helps to illustrate the different behavior of max_u∈U(1,t) g(u, λ, t) with and without attitude constraints. Additionally, it is assumed that the occulter has four thrusters arranged in an equilateral tetrahedral configuration. The alignment of each of these thrusters in the RTN frame in the fixed-attitude mode are given by

\[
U^{\text{thrust}} = \{ \hat{u}_1, \hat{u}_2, \hat{u}_3, \hat{u}_4 \},
\]

\[
\hat{u}_1 = \begin{pmatrix} \sqrt{2/3} \\ 0 \\ -\sqrt{1/3} \end{pmatrix}, \quad \hat{u}_2 = \begin{pmatrix} -\sqrt{2/3} \\ 0 \\ -\sqrt{1/3} \end{pmatrix}, \quad (34)
\]

\[
\hat{u}_3 = \begin{pmatrix} 0 \\ \sqrt{2/3} \\ \sqrt{1/3} \end{pmatrix}, \quad \hat{u}_4 = \begin{pmatrix} 0 \\ -\sqrt{2/3} \\ \sqrt{1/3} \end{pmatrix}
\]

The set U(1, t) for this thruster configuration is illustrated in Figure 5. It is evident from this figure that maneuvers that are nearly aligned with one of the thrusters (corresponding to the vertices of the tetrahedron) are more efficient, while maneuvers that require a combination of thrusters are more expensive because the thrusters partially cancel each other out.

To implement the proposed algorithm, it is necessary to evaluate \( \max_{u ∈ U(1,t)} \lambda^T \Gamma(t) u \) and \( \arg \max_{u ∈ U(1,t)} \lambda^T \Gamma(t) u \). Under the described assumptions, these functions are given in closed-form by

\[
\max_{u ∈ U(1,t)} \lambda^T \Gamma(t) u = \begin{cases} \max_{u ∈ U^{\text{thrust}}} \lambda^T \Gamma(t) u, & 4 < t < 6 \\ \| \Gamma^T(t) \lambda \|_2, & t ≤ 4 \text{ or } t ≥ 6 \end{cases},
\]

\[
\arg \max_{u ∈ U(1,t)} \lambda^T \Gamma(t) u = \begin{cases} \arg \max_{u ∈ U^{\text{thrust}}} \lambda^T \Gamma(t) u, & 4 < t < 6 \\ \frac{\Gamma^T(t) \lambda}{\| \Gamma^T(t) \lambda \|_2}, & t ≤ 4 \text{ or } t ≥ 6 \end{cases}
\]

From this equation, it is evident that the optimal maneuver direction is parallel to \( \Gamma^T(t) \lambda \) when no attitude constraints are enforced. Instead, in the fixed attitude mode the optimal maneuver is to fire the thruster(s) that is closest to parallel to \( \Gamma^T(t) \lambda \).

Key parameters of the solution algorithm are described in the following. For the initialization algorithm, the provided \( T_d \) includes 12 times evenly distributed between \( t_i \) and \( t_f \) and the provided \( \lambda_{\text{init}} \) is a unit vector parallel to the target pseudostate. The initial set of candidate times is selected as the six times in \( T_d \) at which \( \max_{u ∈ U(1,t)} g(u, \lambda_{\text{init}}, t) \) is largest. The tolerances \( \epsilon_{\text{cost}} \) and \( \epsilon_{\text{remove}} \) in the refinement algorithm were selected as 0.01. Finally, the error weight matrix \( Q \) in the optimal control input extraction algorithm is the identity matrix.

The algorithms defined in the previous section were implemented in MATLAB and CVX was used to solve the required convex optimization problems in the iterative refinement and optimal control input extraction algorithms (Grant & Boyd, 2014, 2008).

The proposed algorithm is validated in three different tests. First, the algorithm is used to compute an optimal maneuver sequence for a formation reconfiguration problem representative of the mDOT mission. Second,
a Monte Carlo experiment is performed to demonstrate that the algorithm produces optimal solutions for a wide range of optimal impulsive control problems. The algorithm is initialized with three different sets of candidate times for each test case to characterize the sensitivity of the computation cost to poor initializations. Third, the algorithm is deployed on an embedded microprocessor for nanosatellites to characterize the necessary computation time and demonstrate that the algorithm can be used in real-time applications. In all tested cases, the normalized residual error (||w_{err}||_2/||w||_2), was less than 0.01%, indicating that the solver reliably converged for both the iterative refinement algorithm and the maneuver extraction algorithm.

Example Formation Reconfiguration Problem

The proposed algorithm is first used to compute an optimal maneuver sequence for the formation reconfiguration problem described in the following. The initial mean ROE are computed from the initial mean orbits in Table 2 using [33]. The final mean ROE are selected to ensure that the formation is aligned with Beta Pictoris at the end of the reconfiguration. The initial mean ROE, final mean ROE, and corresponding pseudostate computed from 4 are provided in Table 3. All values in this table are in kilometers. While the ROE change significantly over the reconfiguration (up to 25 km in αδλ), the majority if this change is due to passive dynamics. Indeed, the target pseudostate, which is proportional to the required control effort, is less than 1 km in all state components.

A solution that reaches the target pseudostate and satisfies the optimality criteria to within a tolerance of ε_{cost} was found using only 2 iterations of Algorithm 2. The optimal dual variable is given by \( \lambda_{opt} = 10^{-4} \times [1.18, 0.24, 0.03, -1.04, 0.09, -0.94]^T \). Also, the angle between the optimal dual variable and the pseudostate (arcos(\( \lambda^T w/||\lambda||_2 ||w||_2)) is only 35 degrees, indicating that a vector parallel to \( w \) was a reasonable initial guess for the dual variable. The lower bound on the total delta-v cost computed by evaluating (20) and the total cost of the computed maneuver sequence are both 180.1 mm/s. The optimal maneuver sequence consists of three maneuvers in the RTN frame provided in Table 4. It is noteworthy that these maneuvers include significant radial components, which contradicts the expected behavior from the closed-form solutions proposed in Chernick & D’Amico (2018). This behavior arises from the fact that the reconfiguration must occur in less than one orbit, while Chernick’s closed-form solutions require at least one complete orbit to reconfigure the in-plane ROE (δa, δλ, δε_2, and δε_y).

Table 4: Optimal maneuvers for example scenario.

| t_j (sec) | 0 | 14400 | 36000 |
|------------|----------------|----------------|----------------|
| u_R(t_j) (mm/s) | -47.18 | -15.34 | -78.40 |
| u_T(t_j) (mm/s) | -20.80 | 36.14 | 39.80 |
| u_N(t_j) (mm/s) | -7.06 | -0.26 | 12.49 |

The evolution of \( \max_u U(1,t) g(u, \lambda, t) \) for this solution is illustrated in Figure 6. The optimal maneuver times are indicated by black circles and the time interval in which the fixed attitude constraint is enforced is indicated by gray shading. It is evident from this plot that the optimality criteria are satisfied because \( \max_u U(1,t) g(u, \lambda, t) \leq 1 \) at all times. It is also noteworthy that the time derivative of \( \max_u U(1,t) g(u, \lambda, t) \) is not continuous when the attitude constraint is enforced. This is because \( \max_u U(1,t) g(u, \lambda, t) \) is the maximum of four scalar functions. When the function that defines the maximum changes, this results in a discontinuity in the time derivative.

Monte Carlo Experiment

A Monte Carlo experiment was performed by solving the described example problem for 1000 different target pseudostates. In all of these scenarios the algorithm was able to find a maneuver sequence with a total cost within a factor of ε_{cost} of the lower bound in no more than seven iterations of Algorithm 2. These results demonstrate that the algorithm is able to quickly find optimal solutions for a wide range of impulsive control problems.

To characterize the sensitivity of the computation cost to the initial set of candidate times, the algorithm was initialized for each of these problems with two additional sets of candidate times. The first initial set of times includes

![Figure 6: Evolution of \( \max_u U(1,t) g(u, \lambda, t) \) for optimal solution of example problem including optimal maneuver times (black circles) and attitude constraints (gray).](image-url)
only $t_i$ and $t_f$. This initialization is intended to capture the worst-case computation cost because it is unlikely that the optimal cost can be reached with only two maneuvers. The second initialization includes ten candidate times evenly spaced in the interval $[t_i, t_f]$. This initialization ensures that the initial candidate times are reasonably close to optimal times, but requires the algorithm to check a larger number of constraints in the iterations. The initializations with two, six, and ten candidate times required averages of 3.76, 3.66, and 2.22 iterations of Algorithm 2, respectively. Figure 7 shows the distribution of the number of iterations required to solve these reconfiguration problems for all three initialization schemes. It can be seen that the initializations with two and six candidate times have very similar distributions, suggesting that the algorithm is robust to poor initializations. However, the initialization with 10 times is able to converge in two or three iterations in 90% of the test cases. It follows that initializing the algorithm with more candidate times reduces the number of required iterations. On the other hand, including more candidate times increases the complexity of the optimization problems that must be solved in each iteration. Thus, the ideal number of candidate times for initialization will depend on the limitations of available solvers for a specified application. Overall, these results show that the algorithm is robust to poor initializations and the corresponding increase in the number of required iterations is generally less than a factor of two.

![Figure 7: Distribution of the number of required iterations for reconfiguration problems for three initialization schemes.](image)

**Deployment on an Embedded Microprocessor**

To demonstrate the suitability of this approach for real-time applications, the algorithm was deployed on an embedded microprocessor for nanosatellites. Specifically, the algorithm was deployed on a development board from Tyvak Nanosatellite Systems. The development board, known as a flatsat, is functionally identical to a flight-ready avionics board and includes a microprocessor with an 800 MHz clock speed (Giralo & D’Amico, 2018). To facilitate deployment on this processor, a custom solver produced by CVXGEN (Mattingley & Boyd, 2012) was used to solve the required optimization problems in the iterative refinement and control input extraction algorithms. CVXGEN is an online tool that produces explicit, customized, and efficient solvers for small convex optimization problems that can be represented as quadratic programs. To accommodate the limitations of this solver (only linear constraints), the cost function was changed to the 1-norm of the control input vector for these tests. All other problem specifications are the same as previously described. The algorithm was used to compute optimal maneuver sequences for 12 target pseudostates. These problems required between two and six iterations to reach a solution that satisfies the optimality criteria within a tolerance of $\epsilon_{cost}$ and had total run times ranging from 3.48 to 10.17 seconds. Also, it was found that each iteration required between 1.6 and 1.8 seconds. This behavior was expected because the number of constraints that are evaluated by the solver at each iteration is explicitly coded in the solver. It should be noted that these run times allocate 100% of the CPU power to computation of the optimal maneuver sequences, whereas only a fraction of this power would be available in a real mission with other software. Nevertheless, run times of one minute are still negligible relative to the allowed reconfiguration time of 10 hours. Overall, these results demonstrate that the algorithm can be implemented in embedded applications with run times on the order of seconds. This performance is suitable for a wide range of real-time applications.

### 8. Conclusions

A new solution methodology is proposed for a class of fixed-time, fixed-end-condition linearized optimal impulsive control problems. First, necessary and sufficient optimality conditions are derived for problems with no state constraints where the cost can be expressed as the integral of a time-varying norm-like function of the control input vector. These optimality conditions are leveraged to derive a procedure for quickly computing a lower bound on the minimum cost of a specified problem. Additionally, a three-step algorithm is proposed that provides efficient and robust computation of globally optimal impulsive control input sequences for the described class of optimal control problems. The algorithm accommodates any state definition and dynamics model and can be used with a wide range of cost functions that model the effects of operational constraints on actuation efficiency. The geometry of the problem is leveraged in every step to reduce computational cost and ensure robustness.

The algorithm is validated through implementation in challenging spacecraft formation reconfiguration problems based on the proposed Miniaturized Distributed Occulter/Telescope small satellite mission. This mission consists of a pair of spacecraft in a perturbed, eccentric orbit with time-varying attitude constraints. It is found that the algorithm is able to compute a maneuver sequence with a total cost within 1% of the global optimum within 7 iterations in all test cases. Also, the normalized residual error
of all computed solutions was no larger than 0.01%, indicating reliable convergence. Additionally, the algorithm was deployed on a Tyvak flatsat and it was found that the total run time of this implementation was between 3 and 10 seconds for all test cases. Overall, the proposed solution methodology provides a real-time-capable means of computing globally optimal impulsive control input sequences for a wide range linear time-variant dynamical systems.

Acknowledgments

This work was supported by a NASA Office of the Chief Technologists Space Technology Research Fellowship (NSTRF), NASA Grant #NNX15AP70H and the U.S. Air Force Research Laboratory’s Control, Navigation, and Guidance for Autonomous Spacecraft (CoNGAS) contract FA9453-16-C-0029. The authors would also like to acknowledge Vince Giralo’s assistance in deploying the algorithm on the flatsat.

Appendix A. Dynamics Model

This appendix describes the dynamics model for spacecraft formations in $J_2$-perturbed orbits that is used in the numerical examples in Section 7. Let $\mu$ denote earth’s gravitational parameter, $R_E$ denote earth’s mean radius, and $J_2$ denote earth’s second degree zonal geopotential coefficient. These constants are given by

\[ \mu = 3.986 \times 10^{14} \, m^3/s^2, \quad R_E = 6.378 \times 10^6 \, m, \quad J_2 = 1.082 \times 10^{-3} \]  
(A.1)

The unforced dynamics for the mean orbit elements of a spacecraft in a $J_2$-perturbed orbit are governed by a modified form of Lagrange’s planetary equations (Battin 1987), which are given by

\[ \dot{\mathbf{\alpha}} = \begin{pmatrix} \dot{\alpha} \\ \dot{\epsilon} \\ \dot{i} \\ \dot{\Omega} \\ \dot{\omega} \\ \dot{M} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{3J_2R_E^2\sqrt{\mu}}{2a^{1/2}\eta^4} \cos(i) & 0 & 0 & 0 & 0 \\ \frac{3J_2R_E^2\sqrt{\mu}}{4a^{1/2}\eta^4} (5\cos^2(i) - 1) \end{pmatrix} \]  
(A.2)

It is evident from this equation that $\alpha$, $\epsilon$, and $i$ are constant while $\Omega$, $\omega$, and $M$ vary linearly with time.

Next, it is necessary to model the dynamics of the mean ROE. Using the approach described in Koenig et al. (2017), a STM can be computed by linearizing the equations of motion and solving the linearized system in closed-form. For the ROE definition in [33] and dynamics model in (A.2), the resulting STM is given by

\[ \Phi(\alpha(t_1), \Delta t) = \begin{bmatrix} \Phi_{11} & 0 & 0 & 0 & 0 \\ \Phi_{21} & \Phi_{22} & \Phi_{23} & \Phi_{24} & \Phi_{25} \\ \Phi_{31} & 0 & \Phi_{33} & \Phi_{34} & \Phi_{35} \\ \Phi_{41} & 0 & \Phi_{43} & \Phi_{44} & \Phi_{45} \\ 0 & 0 & 0 & \Phi_{55} & 0 \\ \Phi_{61} & 0 & \Phi_{63} & \Phi_{64} & \Phi_{65} \end{bmatrix} \]  
(A.3)

The nonzero terms of this STM are given by

\[ \Phi_{11} = 1, \quad \Phi_{21} = (-1.5\sqrt{\mu/a^3}\Delta t - 7\kappa\eta P)\Delta t, \]
\[ \Phi_{22} = 1, \quad \Phi_{23} = 7\kappa\epsilon_1 P\Delta t/\eta, \]
\[ \Phi_{24} = 7\kappa\epsilon_1 P\Delta t/\eta, \quad \Phi_{25} = -7\kappa\eta S\Delta t, \]
\[ \Phi_{31} = 3.5\kappa\epsilon_2 Q\Delta t, \]
\[ \Phi_{33} = \cos(\dot{\omega}\Delta t) - 4\kappa\epsilon_1\epsilon_2 GQ\Delta t, \]
\[ \Phi_{34} = -\sin(\dot{\omega}\Delta t) - 4\kappa\epsilon_1\epsilon_2 GQ\Delta t, \]
\[ \Phi_{35} = 5\kappa\epsilon_2 S\Delta t, \quad \Phi_{41} = -3.5\kappa\epsilon_2 Q\Delta t, \]
\[ \Phi_{43} = \sin(\dot{\omega}\Delta t) + 4\kappa\epsilon_1\epsilon_2 GQ\Delta t, \]
\[ \Phi_{44} = \cos(\dot{\omega}\Delta t) + 4\kappa\epsilon_1\epsilon_2 GQ\Delta t, \]
\[ \Phi_{45} = -5\kappa\epsilon_2 S\Delta t, \quad \Phi_{51} = 1, \quad \Phi_{63} = 3.5\kappa S\Delta t, \]
\[ \Phi_{64} = -4\kappa\epsilon_1 GS\Delta t, \quad \Phi_{65} = 2\kappa T\Delta t, \quad \Phi_{66} = 1 \]  
(A.4)

where the substitutions given by

\[ \kappa = \frac{3J_2R_E^2\sqrt{\mu}}{4a^{1/2}\eta^4}, \quad G = \eta^{-2}, \quad P = 3\cos^2(i) - 1, \]
\[ Q = 5\cos^2(i) - 1, \quad S = \sin(2t), \quad T = \sin^2(i), \]
\[ \epsilon_1 = e\cos(\omega(t_1)), \quad \epsilon_2 = e\cos(\omega(t_1)), \]
\[ \epsilon_{x_2} = e\cos(\omega(t_1 + \Delta t)), \quad \epsilon_{y_2} = e\cos(\omega(t_1 + \Delta t)) \]  
(A.5)

are used to simplify notation.

Next, consider the effects of maneuvers. From the Gauss Variational Equations (GVE) (Battin 1987), the change in the osculating orbit elements due to an impulsive maneuvers $u$ performed by a spacecraft in the RTN frame are given by

\[ \Delta \mathbf{\alpha}' = \frac{\partial \mathbf{\alpha}'}{\partial \mathbf{\alpha}} u \]  
(A.6)

under the assumption that the orbit elements are constant over the duration of the maneuver, which is valid for small impulses. Similarly, the change in the mean ROE due to an impulsive maneuver performed by the deputy is given by

\[ \Delta \mathbf{\alpha} = \frac{\partial h(\mathbf{\alpha}_m, \mathbf{\alpha}_d) \partial f_{\text{mean}}(\mathbf{\alpha}_m') \partial \mathbf{\alpha}_d'}{\partial \mathbf{\alpha}_d} \frac{\partial \mathbf{\alpha}_d'}{\partial \mathbf{\alpha}_d} u \]  
(A.7)

where $h$ is the definition of the ROE in [33] and $f_{\text{mean}}$ is the transformation from osculating to mean orbit elements.

To be compatible the algorithm is proposed in this paper, it is necessary to remove the dependency of the control
The input matrix in (A.7) on the relative state. This is done by making three simplifying approximations. The first approximation is given by

\[ \frac{\partial \alpha_c}{\partial \nu_{RTN}} \approx \frac{\partial \alpha_d'}{\partial \nu_{RTN}} \]  
(A.8)

This approximation neglects the effects of the difference between \( \alpha_c \) and \( \alpha_d' \) on the sensitivity of the orbit elements to changes in the velocity of the spacecraft. From the behavior of the GVE (Battin 1987), this approximation is valid for arbitrarily large differences in \( \Omega_d \) and small differences in all other orbit elements. Second, it is evident from the behavior of the analytical transformation in Brouwer (1959) that \( \frac{\partial f_{\text{mean}}(\alpha_d')}{\partial \alpha_d'} \) can be approximated as the identity matrix because all off-diagonal terms are of \( O(J_d) \) or smaller. The third approximation is given by

\[ \frac{\partial h(\alpha_c, \alpha_d)}{\partial \alpha_d} \bigg|_{\alpha_d = \alpha_c} \approx \frac{\partial h(\alpha_c, \alpha_d)}{\partial \alpha_c} \bigg|_{\nu_{RTN}} \]  
(A.9)

This approximation neglects the effects of the separation between the chief and deputy on the sensitivity of the mean ROE to changes in the mean orbit of the deputy. It is evident from the definition of the ROE state in (33) that both second-order partial derivatives with respect to \( \alpha_d, i_d, \Omega_d, \) and \( M_d \) are zero. Thus the approximation is valid for small small differences in \( e \) and \( \nu \) between the chief and deputy and arbitrarily large separations in all other orbit elements. Under these assumptions, the change in the mean ROE due to an impulse applied to the deputy in the RTN frame is given by a control input matrix \( B(\alpha_c) \) that can be computed in closed-form from the mean orbit elements of the chief as given by

\[ B(\alpha_c) = \frac{\Delta x}{\alpha_c, \alpha_d} \frac{\partial h(\alpha_c, \alpha_d)}{\partial \alpha_c} \bigg|_{\alpha_d = \alpha_c} \bigg|_{\nu_{RTN}} \]  
(A.10)

This is identical to the control input matrix used in Chernick & D’Amico (2018), which is given by

\[ B(\alpha) = \sqrt{\frac{\mu}{a_c}} \begin{bmatrix} B_{11} & B_{12} & 0 \\ B_{21} & 0 & 0 \\ B_{31} & B_{32} & B_{33} \\ B_{41} & B_{42} & B_{43} \\ 0 & 0 & B_{53} \\ 0 & 0 & B_{63} \end{bmatrix} \]  
(A.11)

The nonzero terms of this matrix are given by

\[ B_{11} = \frac{2}{\eta} e \sin(\nu), \quad B_{12} = \frac{2}{\eta} (1 + e \cos(\nu)), \]
\[ B_{21} = -\frac{2n^2}{1 + e \cos(\nu)}, \quad B_{31} = \eta \sin(\theta), \]
\[ B_{32} = \frac{\eta (2 + e \cos(\nu)) \cos(\theta) + e \cos(\omega)}{1 + e \cos(\nu)}, \quad B_{33} = \eta \frac{\sin(\omega) \sin(\theta)}{\tan(\theta)(1 + e \cos(\nu))}, \]
\[ B_{41} = -\eta \cos(\theta), \quad B_{42} = \frac{\eta \sin(\omega) \sin(\theta)}{\tan(\theta)(1 + e \cos(\nu))}, \]
\[ B_{43} = \frac{\eta \cos(\omega) \sin(\theta)}{\tan(\theta)(1 + e \cos(\nu))}, \quad B_{53} = \frac{\eta \cos(\theta)}{1 + e \cos(\nu)}, \quad B_{63} = \frac{\eta \sin(\theta)}{1 + e \cos(\nu)} \]

where \( \theta = \omega + \nu \) and \( \nu \) is the true anomaly, which is related to the mean anomaly by Kepler’s equation.

References

Arzelier, D., Bréhard, F., Deak, N., Joldes, M., Louembet, C., Rondipierre, A., & Serra, R. (2016). Linearized impulsive fixed-time fuel-optimal space rendezvous: A new numerical approach. IFAC-PapersOnLine, 49, 373–378.

Battin, R. H. (1987). An Introduction to the Mathematics and Methods of Astrodynamics. New York: AIAA Education Series.

Betts, J. T. (1998). Survey of Numerical Methods for Trajectory Optimization. Journal of Guidance, Control, and Dynamics, 21, 193–207. DOI: 10.2514/2.4231.

Bouchard, B., Dang, N.-M., & Lehalle, C.-A. (2011). Optimal control of Trading Algorithms: A General Impulse Control Approach. SIAM Journal on Financial Mathematics, 2, 404–438. DOI: 10.1137/090777293.

Boyd, S., & Vandenberghe, L. (2004). Convex Optimization. Cambridge University Press.

Brouwer, D. (1959). Solution of the problem of artificial satellite motion without drag. The Astronomical Journal, 64, 378.

Carter, T. E., & Brient, J. (1995). Linearized impulsive rendezvous problem. Journal of Optimization Theory and Applications, 86, 533–584. DOI: 10.1007/BF02192159.

Chernick, M., & D’Amico, S. (2018). New Closed-Form Solutions for Optimal Impulsive Control of Spacecraft Relative Motion. Journal of Guidance, Control, and Dynamics, 41, 301–319. DOI: 10.2514/1.G002848.

D’Amico, S. (2010). Autonomous Formation Flying in Low Earth Orbit. Ph.D. thesis Delft University of Technology.

Gaia, G., D’Amico, S., & Ardaens, J.-S. (2015). Generalised multi-impulsive manoeuvres for optimum spacecraft rendezvous in near-circular orbit. International Journal of Space Science and Engineering, 3, 68–88. DOI: 10.1504/IJSPACESE.2015.069361.

Gilbert, E., & Harasty, G. (1971). A class of fixed-time fuel-optimal impulsive control problems and an efficient algorithm for their solution. IEEE Transactions on Automatic Control, 16, 1–11. DOI: 10.1109/TAC.1971.1099656.

Giralo, V., & D’Amico, S. (2018). Development of the Stanford GNSS Navigation Testbed for Distributed Space Systems. In Institute of Navigation, International Technical Meeting, Reston, Virginia.

Grant, M., & Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, & H. Kimura (Eds.), Recent Advances in Learning and Control Lecture Notes
Grant, M., & Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx.

Handelsman, M., & Lion, P. M. (1968). Primer Vector on Fixed-Time Impulsive Trajectories. AIAA Journal, 6, 127–132.

Jezewski, D. (1980). Primer vector theory applied to the linear relative-motion equations. Optimal control applications and methods, 1, 387–401. DOI: 10.1002/oca.4660010408.

Kim, D.-Y., Woo, B., Park, S.-Y., & Choi, K.-H. (2009). Hybrid optimization for multiple-impulse reconfiguration trajectories of satellite formation flying. Advances in Space Research, 44, 1257–1269. DOI: 10.1016/j.asr.2009.07.029.

Kim, Y. H., & Spencer, D. B. (2002). Optimal Spacecraft Rendezvous Using Genetic Algorithms. Journal of Spacecraft and Rockets, 39, 859–865. DOI: 10.2514/2.3908.

Koenig, A. W., D’Amico, S., Macintosh, B., & Titus, C. J. (2015). Formation design analysis for a miniaturized distributed occulter/telescope in earth orbit. In International Symposium on Space Flight Dynamics (ISSFD), Munich, Germany.

Koenig, A. W., Guffanti, T., & D’Amico, S. (2017). New State Transition Matrices for Spacecraft Relative Motion in Perturbed Orbits. Journal of Guidance, Control, and Dynamics, 40, 1749–1768. DOI: 10.2514/1.G002409.

Lawden, D. F. (1963). Optimal Trajectories for Space Navigation. Butterworths.

Mattingley, J., & Boyd, S. (2012). CVXGEN: A code generator for embedded convex optimization. Optimization and Engineering, 13, 1–27.

Neilson, L. W. (1964). Optimization, a moment problem, and nonlinear programming. Journal of the Society for Industrial and Applied Mathematics, Series A: Control, 2, 33–53. DOI: 10.1137/0302004.

Prussing, J. E. (1969). Optimal Four-Impulse Fixed-Time Rendezvous in the Vicinity of a Circular Orbit. AIAA Journal, 7, 928–935. DOI: 10.2514/3.5246.

Prussing, J. E. (1970). Optimal Two-and Three-Impulse Fixed-Time Rendezvous in the Vicinity of a Circular Orbit. AIAA Journal, 8, 1221–1228. DOI: 10.2514/3.5876.

Prussing, J. E. (1969). Optimal Four-Impulse Fixed-Time Rendezvous in the Vicinity of a Circular Orbit. AIAA Journal, 7, 928–935. DOI: 10.2514/3.5246.

Prussing, J. E. (1970). Optimal Two-and Three-Impulse Fixed-Time Rendezvous in the Vicinity of a Circular Orbit. AIAA Journal, 8, 1221–1228. DOI: 10.2514/3.5876.

Roscoe, C. W. T., Westphal, J. J., Griebach, J. D., & Schaub, H. (2015). Formation Establishment and Reconfiguration Using Differential Elements in J2-Perturbed Orbits. Journal of Guidance, Control, and Dynamics, 38, 1725–1740.

Schaub, H., & Junkins, J. L. (2003). Analytical Mechanics of Space Systems. AIAA.

Serra, R., Arzelier, D., & Rondepierre, A. (2018). Analytical Solutions for Impulsive Elliptic Out-of-Plane Rendezvous Problem via Primer Vector Theory. IEEE Transactions on Control Systems Technology, 26, 207–221. DOI: 10.1109/TCST.2017.2656022.

Sobiesiak, L. A., & Damaren, C. J. (2015). Impulsive Spacecraft Formation Maneuvers with Optimal Firing Times. Journal of Guidance, Control, and Dynamics, 38, 1994–2000. DOI: 10.2514/1.G001095.

Sullivan, J., Grimberg, S., & D’Amico, S. (2017). Comprehensive survey and assessment of spacecraft relative motion dynamics models. Journal of Guidance, Control, and Dynamics, 40, 1837–1859. DOI: 10.2514/1.G002309.

Verriest, E., Delmotte, F., & Egerstedt, M. (2005). Control of epidemics by vaccination. In Proceedings of the 2005 American Control Conference (pp. 985–990). DOI: 10.1109/ACC.2005.1470088.