AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum

10.2514/6.2020-1203

A Real-Time Approach to Minimum-Energy Reorientation of an Asymmetric Rigid Body Spacecraft

Spencer T. McDonald*, Timothy L. Grizzel† Zhenbo Wang‡
The University of Tennessee, Knoxville, Tennessee 37996

The minimum-energy fixed-time rest-to-rest reorientation of an asymmetric rigid-body spacecraft is investigated. The problem is first formulated as a non-convex optimal control problem using a quaternion based dynamic model. Past attempts at solving this problem either sacrifice fidelity to solve analytically or require computationally intensive nonlinear programming (NLP) solvers to tackle the original problem. However, through discretization and constraint implementation, the original optimal control problem is transformed into a sequential convex programming (SCP) problem. Additionally, a simple line search method is introduced to aid in the convergent process of the SCP method. Through numerical demonstrations, the SCP approach has been shown to converge to the minimum-energy optimal solution, even with trivial initial trajectories. The solutions are validated through a comparison with an NLP-based solver, GPOPS. Additionally, line search (LS) has been shown to aid convergence of SCP by decreasing the number of iterations by a significant amount. This introduces a novel approach to solve the minimum-energy fixed-time rest-to-rest reorientation in real time.

Nomenclature

A	=	Jacobian matrix formed by linearized dynamics
B	=	Linear control coefficient matrix
I_x	=	Mass moment of inertia with respect to x-axis
I_y	=	Mass moment of inertia with respect to y-axis
I_z	=	Mass moment of inertia with respect to z-axis
$\mathbf{p}$	=	search direction used in line search
$\mathbf{q}$	=	quaternion state vector
$\mathbf{R}$	=	positive-definite control coefficient matrix
t_0	=	initial time
t_f	=	terminal time
$\mathbf{u}$	=	control vector
$\mathbf{x}$	=	state vector
$\mathbf{z}$	=	state/control optimization matrix
$\hat{\mathbf{z}}$	=	improved state/control matrix from line search
α'	=	optimal step-size found from line search
Δt	=	step time used for numerical integration
δ	=	trust region used in sub-convex QP problem
ϵ	=	convergence criteria used to check for LS-SCP Convergence
η	=	positive coefficient used in line search optimality condition
κ	=	constriction factor used to find α'
$\boldsymbol{\omega}$	=	angular rotation rates state vector
Φ	=	merit function used in line search

*Undergraduate Student, Department of Mechanical, Aerospace, and Biomedical Engineering, AIAA Student Member, smcdon16@vols.utk.edu
†Undergraduate Student, Department of Mechanical, Aerospace, and Biomedical Engineering, AIAA Student Member, tgrizzel@vols.utk.edu
‡Assistant Professor, Department of Mechanical, Aerospace, and Biomedical Engineering, AIAA Member, zwang124@utk.edu

I. Introduction

Spacecraft missions often involve pointing, stabilization, tracking, and other maneuvers that require attitude reorientation of rigid vehicles in free space. As such, there is great interest on how to optimally slew the vehicle into various orientations. Actuation of such maneuvers are carried out by different methods, the most prominent being thrusters, momentum wheels, and rate moment gyros. As previous works have done, this paper will consider control to be linear with respect to each principle axis of the spacecraft, making it ubiquitous and applicable to many actuation schemes. Previous research on this topic is split into three various problems. Those being the following: minimum-time optimization [1, 2], hybrid minimum-time/minimum-energy optimization [3–5], and fixed-time minimum-energy optimization [6]. This work will focus on the latter of such optimization methods. While not as heavily investigated as the minimum-time and hybrid approach, it is clear that the fixed-time minimum-energy reorientation has many significant contributions. Specifically it allows controllable trajectories with final time and terminal conditions known *a priori* to be generated. Additionally, it will serve to minimize energy exerted along that trajectory which aids fuel consumption and improves the feasibility of such maneuvers to be carried out successfully.

In [6], Dixon was one of the first to consider approaches to the fuel-optimal reorientation with large slews required. In [7], modern adaptions of the problem were considered using a more sophisticated and robust quaternion based dynamic model applied to the minimum-fuel problem. Both of which solved the problem analytically using the indirect methods of optimal control theory while assuming the vehicle to be inertially symmetric. More recently, asymmetric body reorientations are tackled using direct optimization methods. In [1], the minimum-time reorientation solution is found using modern NLP solvers. Specifically, Fleming showed many counterintuitive results that arise from the minimum-time reorientation of an asymmetric body as opposed to the previously solved closed form minimum-time symmetric body solution. In [8], they show the ability to use the differential flatness property of an underactuated rigid spacecraft to solve for the minimum-time reorientation with a NLP solver by at most two iterations, allowing them to claim their solution to be real time solvable. In [9], the minimum-time reorientation is solved using the inverse dynamics optimization method, with the ability to again be solved in real time but with necessary trajectories to be computed using a NLP solver offline. As can be seen, the direct method of optimization has increasingly been applied to the spacecraft reorientations; however, many previous works require NLP solvers, which suffer various challenges including unpredictable time consumption and sensitivity to the initial guess.

Throughout recent years, convex optimization has been applied to various engineering problems involving optimal control [10–20]. The advantages of using convex optimization are tremendous. Specifically, there are guarantees that, if a solution exists, it will converge to the globally optimal solution in polynomial time. Additionally, by implementing modern state-of-the-art interior-points method (IPM) solvers, the solution can be found with no initial guess required [21, 22]. Many convex optimization problems are found in the form of either linear programming (LP), quadratic programming (QP), second-order cone programming (SOCP), or semidefinite programming (SDP). In [23], Kim considers the minimum-time problem with specific exclusion zone constraints on the feasible trajectory path of a reorientation maneuver. This is accomplished by introducing convex parameterizations that are equivalent to the nonconvex exclusion constraints. Similar work in [24] was accomplished, where again the problem of minimum-time reorientation in the presence of exclusion zones is explored, with the optimal solution found through sequential semidefinite programming.

In this work, the method of sequential convex programming (SCP) will be implemented to solve the fixed-time minimum-energy reorientation. Like the work in [24], SCP allows NLP problems to be solved by a sequence of convex optimization problems. Unlike the work done in [23, 24], rather than solving a sequence of SDP problems, the work presented here solves a sequence of convex QP problems, which further aids solution convergence and computational efficiency due to QP's lower complexity and more availability of applicable solvers. The fixed-time minimum-energy reorientation is first stated in its original nonconvex general form. After such, the SCP algorithm (composed of convex QP problems) is proposed to find the optimal solution to the nonconvex optimal control problem. Additionally, the SCP method is augmented with a simple line search (LS) routine to improve convergence by penalizing violations from the nonlinear dynamics. To illustrate the effectiveness of the proposed method, two reorientation cases are considered and shown to converge to their optimal solutions. Moreover, solutions are shown to have stable convergence even with trivial or random initial trajectories and line search is shown to reduce the number of iterations by a significant amount.

The rest of the paper is organized as follows. In section II, the quaternion based nonlinear dynamics model along with control constraints will be used to formulate the nonconvex optimal control problem. In section III, the sub-convex QP problem and line search routine will be explained. Additionally, the entire LS-SCP algorithm will be detailed. In section IV, two specific reorientation scenarios will be used to illustrate the effectiveness and robustness of the LS-SCP method at solving for the fixed-time minimum-energy reorientation trajectory.

II. Problem Formulation

A. Equations of Motion

To begin, the reorientation problem will be first presented using the nonlinear equations of motion (EoM). In all, there will be seven state variables and three control variables:

$$\mathbf{x} = [q_1;\ q_2;\ q_3;\ q_4;\ \omega_1;\ \omega_2;\ \omega_3] \in \mathbb{R}^7 \tag{1}$$

$$\mathbf{u} = [u_1;\ u_2;\ u_3] \in \mathbb{R}^3 \tag{2}$$

where $\mathbf{q} = [q_1;\ q_2;\ q_3;\ q_4]$ contains quaternions representing the body orientation of the vehicle in space and $\boldsymbol{\omega} = [\omega_1;\ \omega_2;\ \omega_3]$ represent the body angular rotation rates. The state-space dynamics of the problem are well known and given below [1]:

$$\dot{q_1} = \frac{1}{2}[\omega_1 q_4 - \omega_2 q_3 + \omega_3 q_2] \tag{3}$$

$$\dot{q_2} = \frac{1}{2}[\omega_1 q_3 + \omega_2 q_4 - \omega_3 q_1] \tag{4}$$

$$\dot{q_3} = \frac{1}{2}[-\omega_1 q_2 + \omega_2 q_1 + \omega_3 q_4] \tag{5}$$

$$\dot{q_4} = \frac{1}{2}[-\omega_1 q_1 - \omega_2 q_2 - \omega_3 q_3] \tag{6}$$

$$\dot{\omega_1} = \frac{u_1}{I_x} - \left(\frac{I_z - I_y}{I_x}\right)\omega_2\omega_3 \tag{7}$$

$$\dot{\omega_2} = \frac{u_2}{I_y} - \left(\frac{I_x - I_z}{I_y}\right)\omega_1\omega_3 \tag{8}$$

$$\dot{\omega_3} = \frac{u_3}{I_z} - \left(\frac{I_y - I_x}{I_z}\right)\omega_1\omega_2 \tag{9}$$

where the variables u_1, u_2, and u_3 represent the controls (i.e., applied torques to the system), and the variables I_x,I_y, and I_z are the mass moments of inertia about each respective control axis of the physical body. The quaternion dynamics also share a property of preserving the euclidean norm. As such, assuming a unit quaternion is prescribed as the initial condition then $||\mathbf{q}||_2 = 1$ will hold true throughout the continuous maneuver.

B. Optimal Control Problem

As mentioned before, this paper is investigating the fixed-time minimum-energy reorientation problem. As such, the final time (t_f) of the reorientation maneuver is assumed to be known *a priori*, along with the terminal constraints. The maneuver will also be assumed to be rest-to-rest, requiring the initial and terminal rotation rates to be zero. Due to the nature of quaternions, the initial orientation can always be assumed nominal (i.e., unrotated) as any maneuver can be described as a rotation from its current position. Additionally, constraints on both state and control need to be considered to ensure feasibility of the maneuver and ensure safe operating conditions for the vehicle. With these considerations, the optimal control problem can be formulated below:

$$\underset{\mathbf{x},\mathbf{u}}{\text{Minimize:}} \quad J = \int_{t_0}^{t_f} \mathbf{u}^T(t) R\, \mathbf{u}(t)\, dt \tag{10}$$

$$\text{Subject to:} \quad \dot{\boldsymbol{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{11}$$

$$\mathbf{q}(t_0) = [0;\ 0;\ 0;\ 1] \tag{12}$$

$$\boldsymbol{\omega}(t_0) = \mathbf{0} \tag{13}$$

$$\mathbf{q}(t_f) = [q_1(t_f);\ q_2(t_f);\ q_3(t_f);\ q_4(t_f)] \tag{14}$$

$$\boldsymbol{\omega}(t_f) = \mathbf{0} \tag{15}$$

$$\mathbf{x}_{min} \le \mathbf{x}(t) \le \mathbf{x}_{max} \tag{16}$$

$$\mathbf{u}_{min} \le \mathbf{u}(t) \le \mathbf{u}_{max} \tag{17}$$

where R is a positive definite matrix (assumed as the identity matrix in this paper) used to generate a quadratic cost function in an effort to minimize control exertion. It is worth mentioning that due to the highly nonlinear dynamics and asymmetric interitas, the problem is extremely hard to solve analytically. Moreover, many NLP numeric solvers have difficulty converging to an optimal solution in real time.

III. Methodology

As can be seen, the previous optimal control problem is nonconvex due to the highly nonlinear nature of the dynamics which include coupled state and control. As such, analytical solutions may require sacrificing fidelity (e.g., assuming axisymmetric, simplified boundary conditions, etc.) or could be hard to impossible to solve in general. Because of this, the only applicable solver for such problems are NLP solvers. However, many problems may be encountered when implementing NLP solvers. For example, most off-the-shelf NLP solvers have no guarantees that a solution will converge or, if it does, that it will require non-polynomial time computations.

Of particular interest in recent years is the use of convex optimization in solving optimal control problems. If an optimal control problem can be reformulated into a convex form, there are efficient algorithms that can solve the problems in polynomial time with guaranteed optimality and convergence. Therefore, this section will show how to convert the previous nonconvex optimal control problem into a sequence of convex QP problems. This will be accomplished by the following: first, the dynamics are linearized through a Jacobian expansion; second, the continuous problem is discretized and the dynamics are enforced through trapezoidal integration; third, final constraints are employed and the problem is formulated as a sub-convex QP problem; fourth, line search is implemented to penalize deviations from true dynamics; and fifth, the LS-SCP algorithm is detailed.

A. Linearization

Before the problem can be tackled by a sequence of convex optimization sub problems, the nonlinear dynamics must be presented in a convex form. To this end, the dynamics are enforced by way of affine equality constraints. Before such constraints can be implemented, the dynamics are linearized by way of Jacobian expansion. The control and state variables as seen in the EoM already share the property that they are decoupled. As such, the equations of motion can be written in the standard linear control form below:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + B\mathbf{u} \tag{18}$$

with

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{1}{2}[\omega_1 q_4 - \omega_2 q_3 + \omega_3 q_2] \\ \frac{1}{2}[\omega_1 q_3 + \omega_2 q_4 - \omega_3 q_1] \\ \frac{1}{2}[-\omega_1 q_2 + \omega_2 q_1 + \omega_3 q_4] \\ \frac{1}{2}[-\omega_1 q_1 - \omega_2 q_2 - \omega_3 q_3] \\ -(\frac{I_z - I_y}{I_x})\omega_2\omega_3 \\ -(\frac{I_x - I_z}{I_y})\omega_1\omega_3 \\ -(\frac{I_y - I_x}{I_z})\omega_1\omega_2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{I_x} & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix} \tag{19}$$

The Jacobian matrix A can then be formed from the nonlinear state dynamics, $\mathbf{f}(\mathbf{x})$:

$$A(\mathbf{x}_*) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\mathbf{x}_*} = \begin{bmatrix} 0 & \frac{\omega_3}{2} & -\frac{\omega_2}{2} & \frac{\omega_1}{2} & \frac{q_4}{2} & -\frac{q_3}{2} & \frac{q_2}{2} \\ -\frac{\omega_3}{2} & 0 & \frac{\omega_1}{2} & \frac{\omega_2}{2} & \frac{q_3}{2} & \frac{q_4}{2} & -\frac{q_1}{2} \\ \frac{\omega_2}{2} & -\frac{\omega_1}{2} & 0 & \frac{\omega_3}{2} & -\frac{q_2}{2} & \frac{q_1}{2} & \frac{q_4}{2} \\ -\frac{\omega_1}{2} & -\frac{\omega_2}{2} & -\frac{\omega_3}{2} & 0 & -\frac{q_1}{2} & -\frac{q_2}{2} & -\frac{q_3}{2} \\ 0 & 0 & 0 & 0 & 0 & (\frac{I_y - I_z}{I_x})\omega_3 & (\frac{I_y - I_z}{I_x})\omega_2 \\ 0 & 0 & 0 & 0 & -(\frac{I_x - I_z}{I_y})\omega_3 & 0 & -(\frac{I_x - I_z}{I_y})\omega_1 \\ 0 & 0 & 0 & 0 & (\frac{I_x - I_y}{I_z})\omega_2 & (\frac{I_x - I_y}{I_z})\omega_1 & 0 \end{bmatrix}_{\mathbf{x}=\mathbf{x}_*} \tag{20}$$

Using this expansion, first order approximations can be generated using the equation below. This will be used along with a known prior state history ($\mathbf{x}_*$) to enforce the approximate dynamics throughout the SCP method.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}_*) + A(\mathbf{x}_*)(\mathbf{x} - \mathbf{x}_*) + B\mathbf{u} \tag{21}$$

B. Discretization

To solve this problem numerically, the continuous dynamics must be transferred into a discrete optimization problem. To do so, the time interval $(t_0 - t_f)$ is divided into N evenly spaced nodes $([t_1, t_2, \dots, t_N])$ such that $t_1 = t_0$ and $t_N = t_f$ with $\Delta t = t_2 - t_1$. The state and control vector are then discretized by the same number of nodes, such that $\mathbf{x}(t_i)$ and $\mathbf{u}(t_i)$.

To enforce the dynamics, trapezoidal integration will be used. As mentioned in the previous section, a previous state history will be used to approximate the dynamics of the new iteration. As such, the convention is in $\mathbf{x}_i^k$, k will represent the current state iteration, and i will represent the current node being evaluated. The trapezoidal integration can then be written in this form below:

$$\mathbf{x}_i^{k+1} = \mathbf{x}_{i-1}^{k+1} + \frac{\Delta t}{2}((\mathbf{A}_{i-1}^k \mathbf{x}_{i-1}^{k+1} + \mathbf{B}_{i-1}^k \mathbf{u}_{i-1}^{k+1} + \mathbf{b}_{i-1}^k) + (\mathbf{A}_i^k \mathbf{x}_i^{k+1} + \mathbf{B}_i^k \mathbf{u}_i^{k+1} + \mathbf{b}_i^k)) \tag{22}$$

where kth iteration is presumably known and the $k+1$ is the iteration currently being solved. Additionally, $\mathbf{b}_i^k = \mathbf{f}(\mathbf{x}_i^k, \mathbf{u}_i^k)$, which is the dynamics evaluated at the kth iteration and the ith node. This trapezoidal integration can be further simplified into a single constraint as such:

$$H_{i-1}^k \mathbf{x}_{i-1}^{k+1} - H_i^k \mathbf{x}_i^{k+1} + G_{i-1}^k \mathbf{u}_{i-1}^{k+1} + G_i^k \mathbf{u}_i^{k+1} = -\frac{\Delta t}{2}(\mathbf{b}_{i-1}^k + \mathbf{b}_i^k) \tag{23}$$

with

$$H_{i-1}^k = I + \frac{\Delta t}{2}\mathbf{A}_{i-1}^k \tag{24}$$

$$H_i^k = I - \frac{\Delta t}{2}\mathbf{A}_i^k \tag{25}$$

$$G_{i-1}^k = \frac{\Delta t}{2}\mathbf{B}_{i-1}^k \tag{26}$$

$$G_i^k = \frac{\Delta t}{2}\mathbf{B}_i^k \tag{27}$$

where I equals the identity matrix. This form is easily implemented into a convex solver and ensures that the first order approximations are satisfied through the trapezoidal integration.

A valid concern is whether or not $||\mathbf{q}||_2 = 1$ is maintained throughout this discretization. More precisely, will the SCP method maintain unit quaternions. While it is true that the dynamics ensure the quaternion to be normalized throughout the maneuver, the first order approximation makes no such guarantees. However, in Section IV, numerical demonstrations will show that the convergent SCP solution can make $||\mathbf{q}||_2$ arbitrarily close to one, meaning that the convergent solution does indeed hold true that $||\mathbf{q}||_2 = 1$ is maintained.

As another matter of convention, the a combined state and control matrix $\mathbf{z}^k \in \mathbb{R}^{10\times N}$ where $\mathbf{z}^k = [\mathbf{x}^k; \mathbf{u}^k]$ will contain all discretized values. This is the optimization matrix of unknown values for which each sequential convex optimization problem will solve.

C. Sequential Convex Programming

To begin the SCP method, the nonconex optimal control problem must be formatted and appropriately stated as a convex optimization problem so that it can be solved by an applicable solver. Additionally, an initial trajectory, $\mathbf{z}^0$, is assumed to have been known before the SCP method begins. Discussion on how to provide such initial trajectories can be found in Section IV. The first step is to format the objective as a discrete optimization problem rather than the continuous functional described in (10). The time integral is approximated as follows:

$$J = \Delta t \sum_{n=1}^{N} ||\mathbf{u}_i^{k+1}||_2^2 \tag{28}$$

It is clear to see the equivalence between (28) and (10). It is also clear that (28) forms a convex quadratic objective, of which a global minimum could exist. Additionally, lower and upper bounds are added to the state and control to ensure safety, feasibility, and to lessen the search space of the solver. The bounds for the problem are the following:

$$||\mathbf{q}||_\infty \leq 1 \tag{29}$$

$$||\boldsymbol{\omega}||_\infty \leq \boldsymbol{\omega}_{max} \tag{30}$$

$$||\mathbf{u}||_\infty \leq \mathbf{u}_{max} \tag{31}$$

where $\boldsymbol{\omega}_{max}$ is used as the upper limit of angular slew to ensure structural safety of the vehicle and $\mathbf{u}_{max}$ is implemented as the maximum control capability of the vehicle. Ensuring that the max quaternion does not exceed one helps lower the search path which speeds convergence, but otherwise is not entirely necessary. It is important to note that while the L_∞ norm is currently being used for all state and control constraints, any norm could be put in its place and still maintain a convex optimization problem. The advantage of using the L_∞ norm is that it renders the problem being considered as a convex QP program which is inherently simpler than many other optimization programs. The same could be said about the objective function as well. Presently, the convex quadratic control objective is considered. However, any convex function could be used as the objective and the SCP method could be applied to solve it.

Along the same lines, to help encourage convergence, a trust region $\delta \in \mathbb{R}^7$ where $\delta > 0$ is enforced on the state iterations. This is to ensure that the first order approximation does not stray away too far from the actual dynamic values. This is done by the following:

$$|\,\mathbf{x}_i^{k+1} - \mathbf{x}_i^k| \leq \delta \tag{32}$$

With all these considerations, the convex quadratic program can be considered as the following:

$$\underset{\mathbf{x},\mathbf{u}}{\text{Minimize:}} \quad J = \Delta t \sum_{n=1}^{N} \|\mathbf{u}_n^{k+1}\|_2^2 \tag{33}$$

$$\text{Subject to:} \tag{34}$$

$$H_{i-1}^k \mathbf{x}_{i-1}^{k+1} - H_i^k \mathbf{x}_i^{k+1} + G_{i-1}^k \mathbf{u}_{i-1}^{k+1} + G_i^k \mathbf{u}_i^{k+1} = -\frac{\Delta t}{2}(\mathbf{b}_{i-1}^k + \mathbf{b}_i^k) \quad \forall i \in (2,3,...,N) \tag{35}$$

$$\|\mathbf{q}\|_\infty \leq 1 \tag{36}$$

$$\|\boldsymbol{\omega}\|_\infty \leq \boldsymbol{\omega}_{max} \tag{37}$$

$$\|\mathbf{u}\|_\infty \leq \mathbf{u}_{max} \tag{38}$$

$$|\,\mathbf{x}_i^{k+1} - \mathbf{x}_i^k| \leq \delta \quad \forall i \in (1,2,...,N) \tag{39}$$

where, again, the state/control matrix $\mathbf{z}^k$ is assumed to be known before the $k+1$ iteration. Each convex QP problem will indeed be solved in polynomial time and converge to the globally optimal solution. If the SCP method works correctly, the iterations will converge to an optimal solution of the original problem. As such, the difference between iterations will decrease. The test used for convergence is the following. A small $\epsilon \in \mathbb{R}^7$ such that $\epsilon > 0$ and $\epsilon < \delta$ is used as the "convergence criteria". The SCP method is said to converge when the following condition is satisfied:

$$\max_i |\,\mathbf{x}_i^{k+1} - \mathbf{x}_i^k| < \epsilon \tag{40}$$

D. Line Search

There has been recent success in using a simple line search routine in conjunction with SCP. Line search is an algorithm that finds an pseudo-optimal step length (α) and search direction ($\mathbf{p}$) in an effort to further minimize some cost function. For the present paper, line search will be used to "close the gap" between the true nonlinear dynamics and the first order approximation of the dynamics. As the SCP method only converges when the original dynamics are satisfied, when implemented correctly, this could drastically improve convergence to the optimal solution, the results of which will be discussed in Section IV.

The line search method used here is based off the work done in [12, 25]. After using line search to find (α) and ($\mathbf{p}$), the updated state/control matrix is given by:

$$\hat{\mathbf{z}}^{k+1} = \mathbf{z}^{k+1} + \alpha\mathbf{p} \tag{41}$$

where $\mathbf{z}^{k+1}$ would be the optimization matrix found from the SCP method and $\hat{\mathbf{z}}^{k+1}$ would be the "improved" value found from line search. In the above $\mathbf{p} \in \mathbb{R}^{10\times N}$ is considered the step direction and $\alpha \in (0,\, 1]$ is considered the step size. The value of $\mathbf{p}$ is set to be the difference between the SCP iterations:

$$\mathbf{p} = \mathbf{z}^{k+1} - \mathbf{z}^k \tag{42}$$

From there, the line search algorithm is used to find the best value of α to satisfy some optimally conditions. As mentioned before, the purpose of line search is to "close the gap" between nonlinear dynamics and the approximate dynamics. To do so, one must create a cost function that can appropriately quantify the deviation from the nonlinear dynamics. The merit function of choice is defined by the following:

$$\Phi(\mathbf{z}^k) = \sum_{n=2}^{N} \|g(\mathbf{z}_n^k)\|_1 \tag{43}$$

where

$$g(\mathbf{z}_i^k) = [\mathbf{x}_{i-1}^k + \frac{\Delta t}{2}(\mathbf{f}(\mathbf{x}_{i-1}^k, \mathbf{u}_{i-1}^k) + \mathbf{f}(\mathbf{x}_i^k, \mathbf{u}_i^k))] - \mathbf{x}_i^k \tag{44}$$

It is easy to see that $[\mathbf{x}_{i-1}^k + \frac{\Delta t}{2}(\mathbf{f}(\mathbf{x}_{i-1}^k, \mathbf{u}_{i-1}^k) + \mathbf{f}(\mathbf{x}_i^k, \mathbf{u}_i^k))]$ should be closer to the true value of $\mathbf{x}_i^k$ as it is trapezoidal integration of the true dynamics, not the approximated dynamics. As such, the function $g(\mathbf{x}_i^k)\colon \mathbb{R}^{10} \to \mathbb{R}^7$, can be thought of as finding the deviation between the true dynamics and the approximated dynamics for each state. Thus, it is clear that the function $\Phi(\mathbf{z}^k)\colon \mathbb{R}^{10\times N} \to \mathbb{R}$ is a cost function of how much the values deviate from the approximate dynamics. Line search will seek to find the largest step size (α) that also aims at lowering the cost function $\Phi(\mathbf{z}^k)$. The process of line search starts by initially setting $\alpha = 1$. Next, the following condition is examined to see if it is satisfied for some given $\eta \in (0, 1)$:

$$\Phi(\mathbf{z}^k + \alpha\mathbf{p}) \leq \Phi(\mathbf{z}^k) + \eta\alpha D(\Phi(\mathbf{z}^k), \mathbf{p}) \tag{45}$$

where $D(\Phi(\mathbf{z}^k), \mathbf{p})$ is the directional derivative of the function $\Phi(\mathbf{z}^k)$ as it moves in the search direction $\mathbf{p}$. The value of $D(\Phi(\mathbf{z}^k), \mathbf{p})$ can be very accurately approximated by using an extremely small ϵ' with this equation:

$$D(\Phi(\mathbf{z}^k), \mathbf{p}) \approx \frac{\Phi(\mathbf{z}^k + \epsilon'\mathbf{p}) - \Phi(\mathbf{z}^k)}{\epsilon'} \tag{46}$$

If $\alpha = 1$, then $\alpha' = \alpha$, where α' is considered the optimum step size from line search. If $\alpha = 1$ does not satisfy the condition in (45), the next step is to use a so called constriction factor, $\kappa \in (0, 1)$. The process is iterative so that α is updated by $\alpha = \kappa\alpha$ until it is satisfied for some α'. Thus it is clear that $\kappa\alpha \to \alpha'$. After α' is found, the optimum $\hat{\mathbf{z}}^{k+1}$ can be found from (41). From there, the method sets $\mathbf{z}^{k+1} = \hat{\mathbf{z}}^{k+1}$ so that $\mathbf{z}^{k+1}$ can be used as the previous trajectory for the next SCP iteration. Choosing a given κ and η is subjective and problem specific. In Section IV, the process of choosing κ and η will be discussed for the problem currently being investigated.

E. LS-SCP Algorithm

The proposed LS-based SCP (LS-SCP) process is summarized as follows:

1) Set $k = 0$. Tthe initial trajectory ($\mathbf{z}^0$) is assumed to be known or generated *a priori*. Moreover, It will be shown later on that simple linear interpolation between initial and terminal conditions will suffice for this method. Additionally, set positive values of κ and η to be used for line search.

2) Use $\mathbf{z}^k$ and solve for the $\mathbf{z}^{k+1}$ by way of the convex QP problem laid out in (33)-(39).

3) After generating the $\mathbf{z}^{k+1}$ trajectory, the convergence criteria in (40) is checked. If the convergence criteria is met, then the optimal trajectory is given as $\mathbf{z}^{k+1}$. If not, move to the next step.

4) Use the simple line search method spelled out in the previous subsection to find the optimum step size α' and search direction $\mathbf{p}$. Using those values, set $\mathbf{z}^{k+1} = \mathbf{z}^k + \alpha'\mathbf{p}$. After such, set $k = k + 1$ and move to step (2).

IV. Numerical Demonstrations

This section presents the preliminary data found using the LS-SCP method laid out in the previous sections. To illustrate the abilities of the LS-SCP method, two reorientation cases are considered. The first is a baseline case just to show the effectiveness of the developed method by considering a small angle slew maneuver. Specifically, it will show that LS aids convergence by decreasing the number of iterations from 11 to 7. Additionally, the final trajectory is compared with an NLP-based commercial optimal control solver (GPOPS) and is found to match exactly with the optimal solution. The second case considered is a large slew maneuver, but this time the initial trajectories are found by sampling from a normal distribution with a standard deviation of 0.01. The SCP method is found to convergence even in the face of this random initial trajectory. This illustrates the robustness of the LS-SCP method with respect to initial trajectories. Finally, the converged trajectory of case 2 is compared with GPOPS and, again, found to be the optimal solution to the original problem.

In the simulations, the state and control variables are discretized into 100 nodes. For the purpose of numerical demonstrations, the rigid body intertias and parameters are taken from publicly available data on NASA's X-Ray Timing Explorer (XTE) vehicle (Fig. 1). Each iteration is solved for by carrying out the LS-SCP algorithm laid out in the previous section. The convergence and trust region are given as $\epsilon = 10^{-5}$ and $\delta = 0.1$. All iterations are seen from a dark blue to bright red, with dark blue being the initial trajectory and bright red being the final, converged trajectory. The problem formulation and the method are implemented in ECOS, which is a state-of-the-art convex solver that applies a primal-dual predictor-corrector method along with the Nesterov-Todd scaling and self-dual embedding techniques [26].

Fig. 1 NASA's X-Ray Timing Explorer (XTE) vehicle

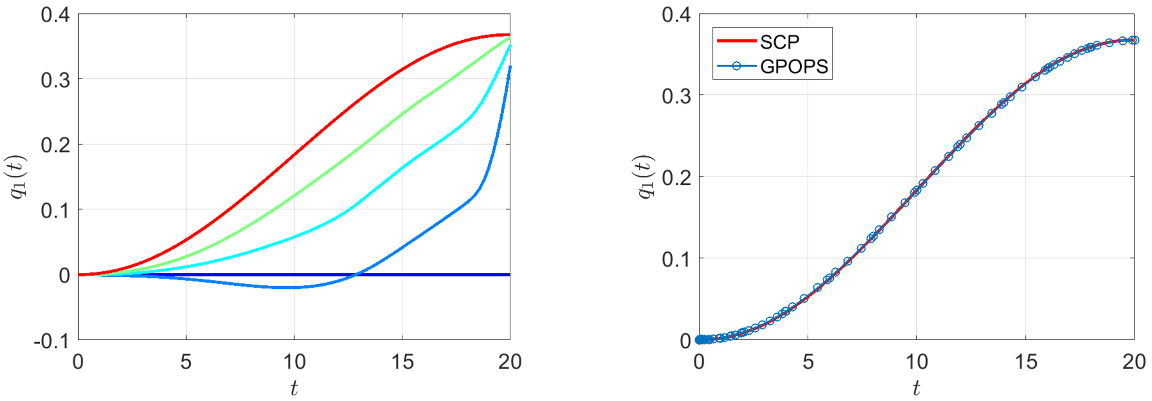
Table 1 NASA XTE parameters used in simulations

Parameter	Value	Units
I_x	5621	$kg \times m^3$
I_y	4547	$kg \times m^3$
I_z	2364	$kg \times m^3$
$\boldsymbol{\omega}_{max}$	1	rad/s
$\mathbf{u}_{max}$	50	$N \times m$

A. Convergence and Merits of Line Search

As mentioned before, the first case considered is a relatively small angle slew maneuver. The problem currently considered is a rest-to-rest rotation with the terminal angle found from the Euler angle rotation sequence XYZ. The arbitrary values of the rotation sequence are set to be [40, 20, 30] deg. The terminal time (t_f) is set as 20 seconds. The LS values are set as $\kappa = 0.9$ and $\eta = 0.7$, after a quick analysis showed this combination worked quite well. The initial trajectory ($\mathbf{z}^0$) is found by setting state and control variables to zero with the exception of setting $\boldsymbol{\omega} = 0.1$ as this was found to promote convergence. In all the figures below, (a) will present all the iterations of the LS-SCP method to illustrate how it converges while (b) will show the final trajectory comparison to the NLP solver GPOPS.

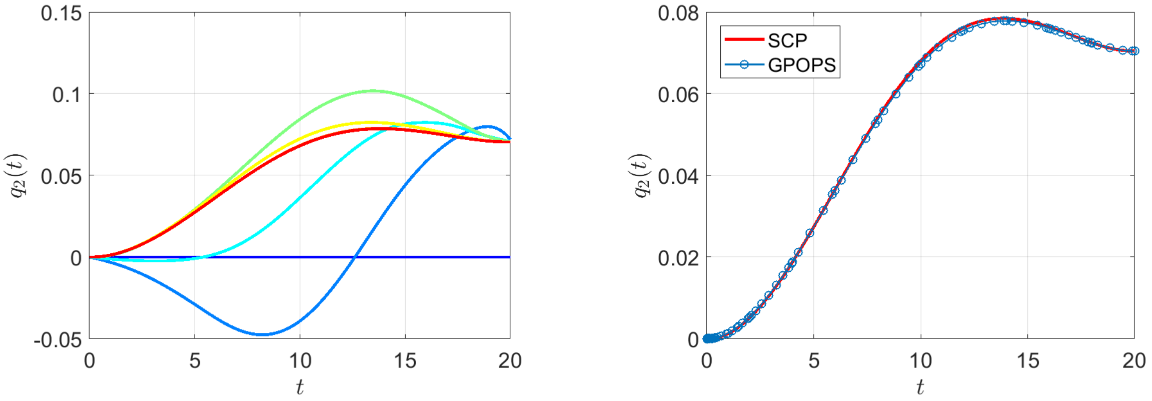
It was found that with the LS-SCP method, a solution converged in 7 iterations. However, when running this method with no LS, it took 11 iterations in all. This means that LS alone accounted for reducing a substantial amount of the total iterations to find a convergent solution. Moreover, in (b) it is quite easy to verify the optimality of the converged trajectory as they all pass straight through the optimal GPOPS solution found from the NLP solver. It is also important to note, that the max error of the norm quaternion was less than 10^{-5} throughout the entire maneuver. This shows that the converged trajectory found through the LS-SCP method preserved the unit quaternion with a negligible error. This again ensures that dynamics are satisfied and speaks to the convergent ability of the LS-SCP method.



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

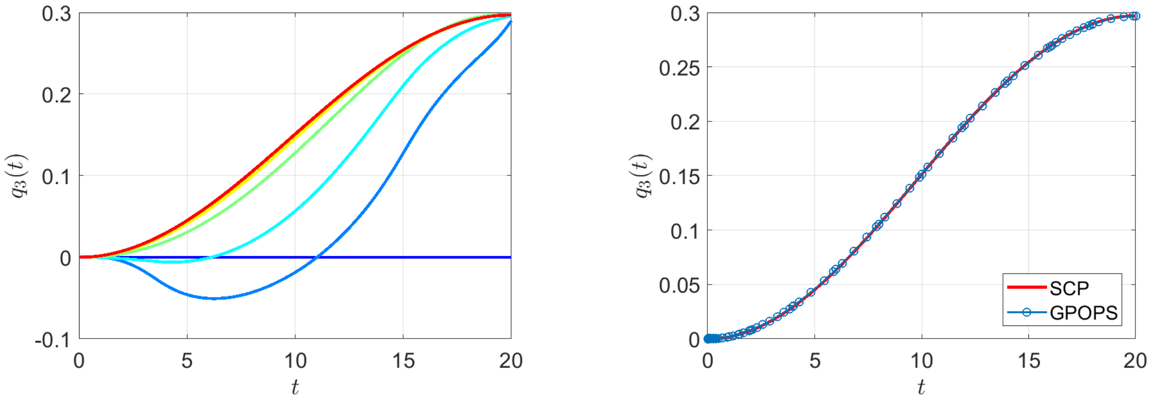
Fig. 2 Case 1: Convergence and Comparison of q_1



(a) LS-SCP Convergence

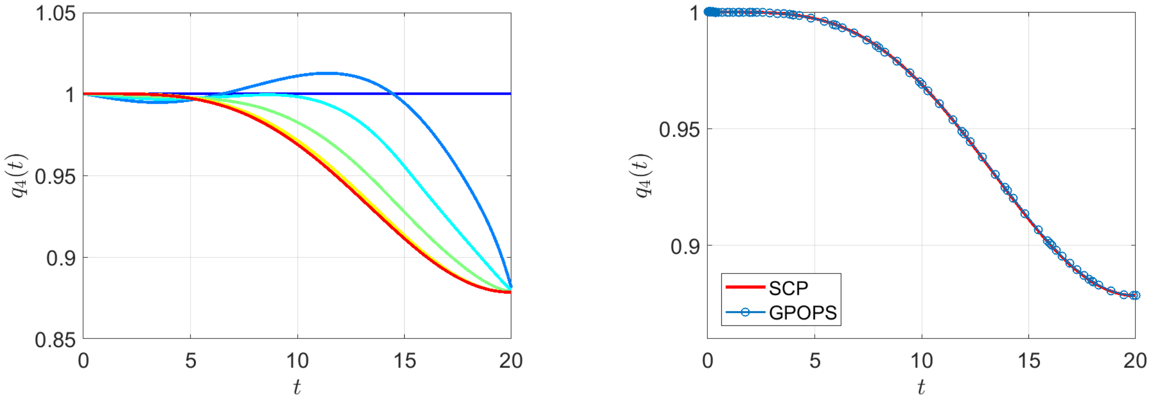
(b) LS-SCP Comparison with GPOPS

Fig. 3 Case 1: Convergence and Comparison of q_2



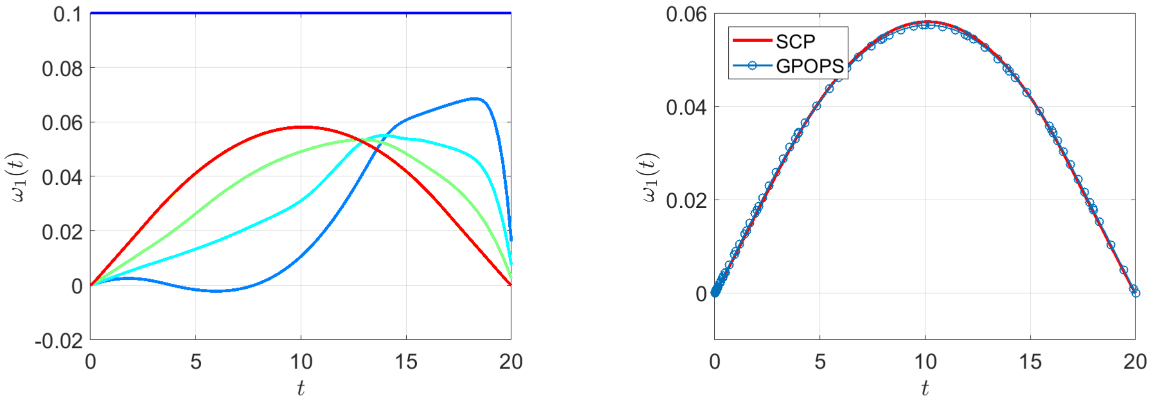
(a) LS-SCP Convergence **(b) LS-SCP Comparison with GPOPS**

Fig. 4 Case 1: Convergence and Comparison of q_3



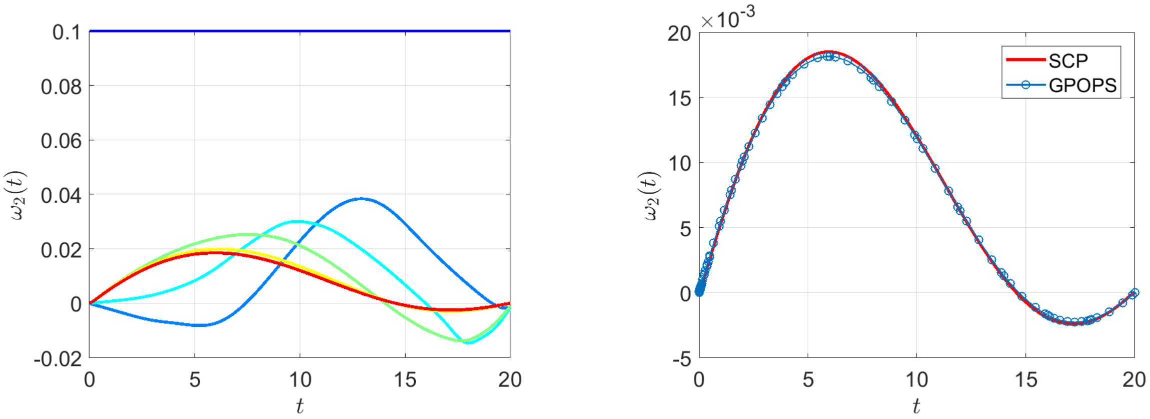
(a) LS-SCP Convergence **(b) LS-SCP Comparison with GPOPS**

Fig. 5 Case 1: Convergence and Comparison of q_4



(a) LS-SCP Convergence **(b) LS-SCP Comparison with GPOPS**

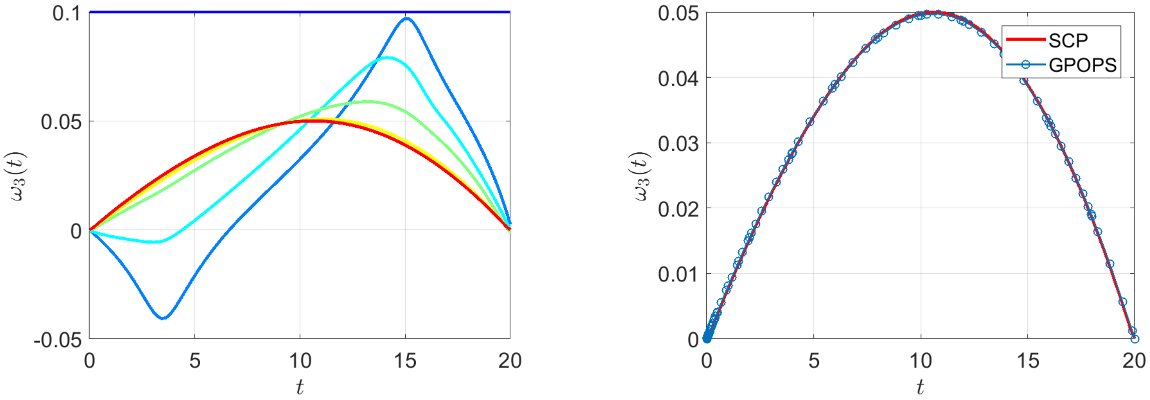
Fig. 6 Case 1: Convergence and Comparison of ω_1



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

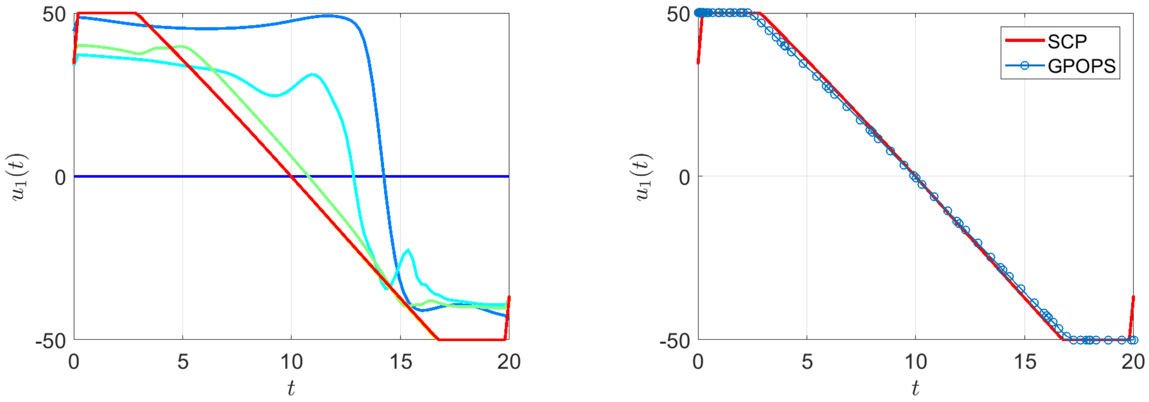
Fig. 7 Case 1: Convergence and Comparison of ω_2



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

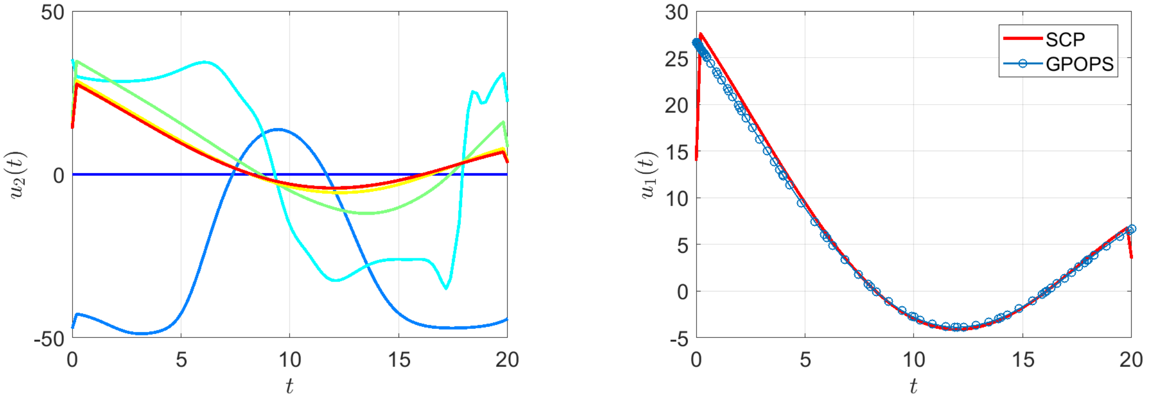
Fig. 8 Case 1: Convergence and Comparison of ω_3



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

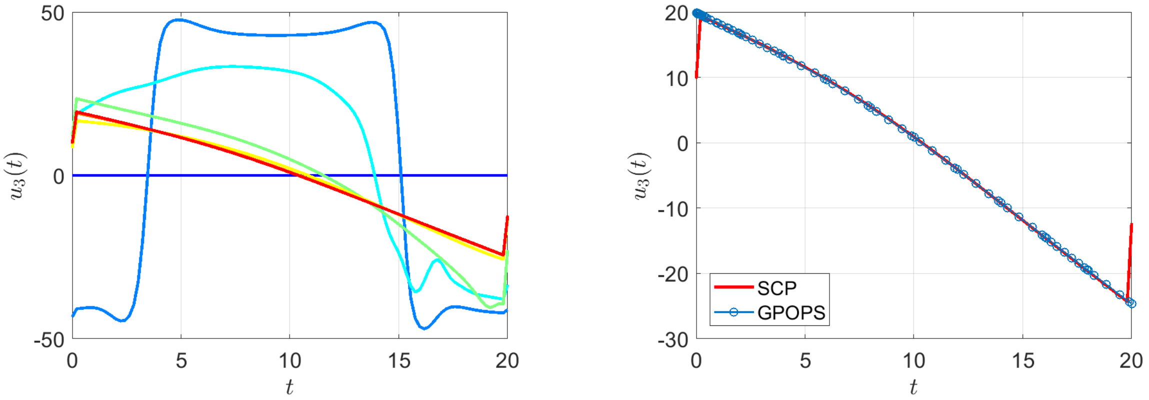
Fig. 9 Case 1: Convergence and Comparison of u_1



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

Fig. 10 Case 1: Convergence and Comparison of u_2



(a) LS-SCP Convergence

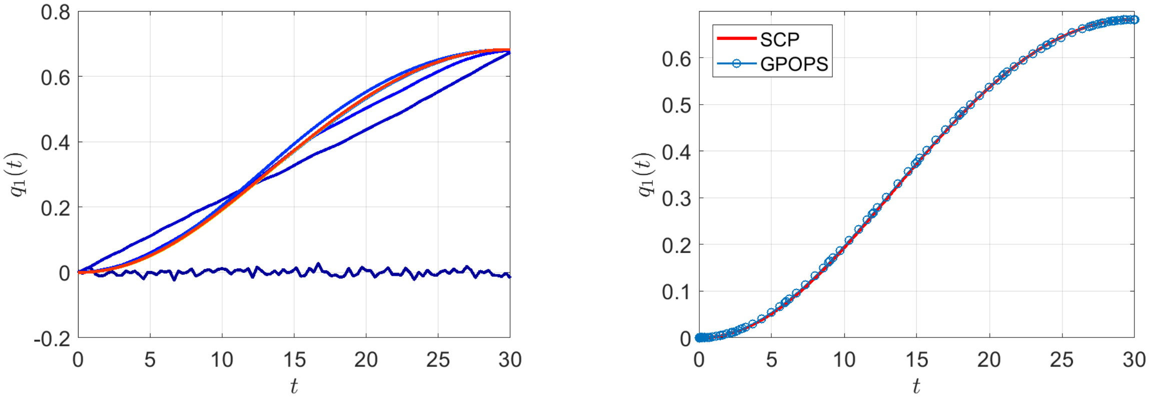
(b) LS-SCP Comparison with GPOPS

Fig. 11 Case 1: Convergence and Comparison of u_3

B. Robustness of LS-SCP Method

In this subsection, the second case is considered with a large slew maneuver. The problem, again, is a rest-to-rest rotation with the terminal angle found from the Euler angle rotation sequence XYZ. This time, the values of the rotation sequence are set to be [100, 20, 70] deg. The terminal time of this maneuver is set at 40 seconds (t_f). The aforementioned LS parameters are again set as $\kappa = 0.9$ and $\eta = 0.7$. The biggest change for case 2 is how the initial trajectory ($\mathbf{z}^0$) is found. To illustrate the robustness of the LS-SCP method presented in this paper, ($\mathbf{z}^0$) is found by sampling from a random zero-mean Gaussian distribution with a standard deviation of 0.01. Like in case 1, (a) will present all the iterations of the LS-SCP method to illustrate how it converges while (b) will show the final trajectory comparison to the NLP solver GPOPS.

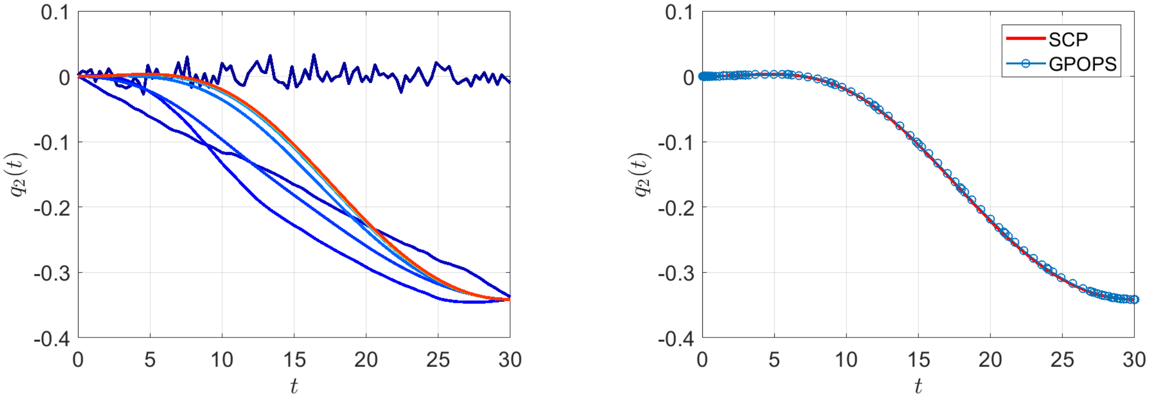
The first observation is again that the LS-SCP method does indeed converge even when exposed to this highly perturbed initial trajectory with 16 iterations in total. This speaks to the robustness of the LS-SCP method with respect to various initial trajectories. The convergent solution is aided by LS, which promotes the SCP method to output more feasible trajectories with each iteration. As in case 1, the solutions clearly match the optimal solutions found using GPOPS. Overall, this presents strong evidence to favor this method when solving the minimum-energy fixed-time reorientation problem.



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

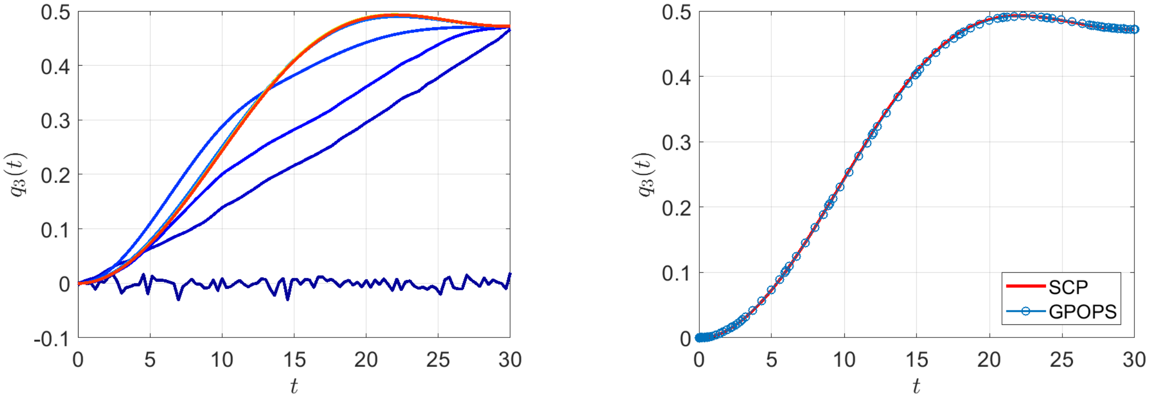
Fig. 12 Case 2: Convergence and Comparison of q_1



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

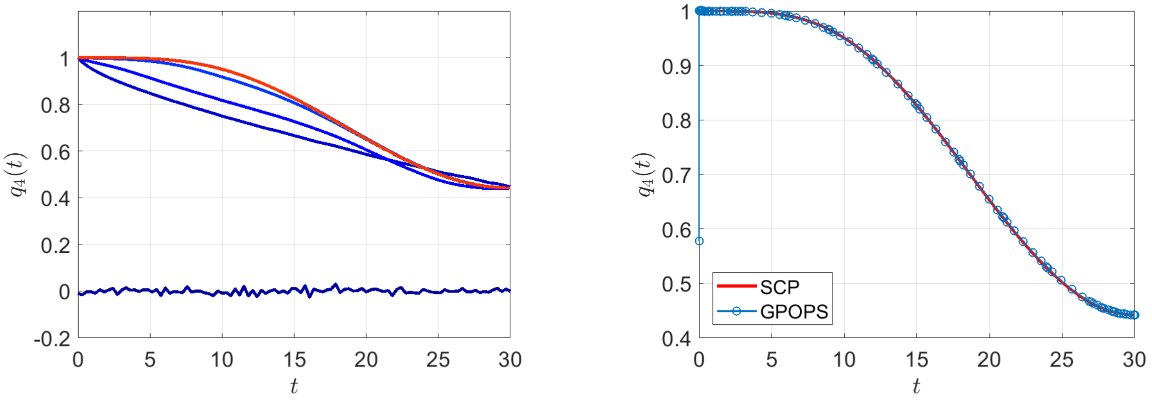
Fig. 13 Case 2: Convergence and Comparison of q_2



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

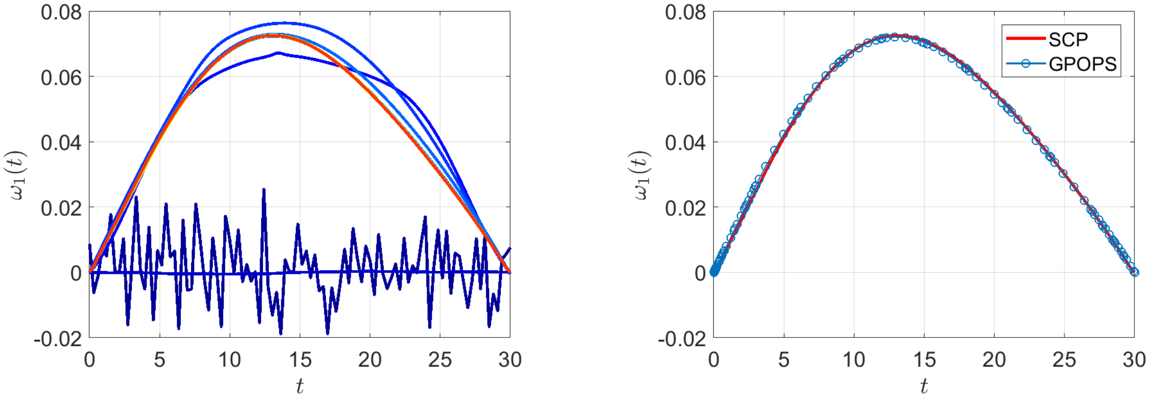
Fig. 14 Case 2: Convergence and Comparison of q_3



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

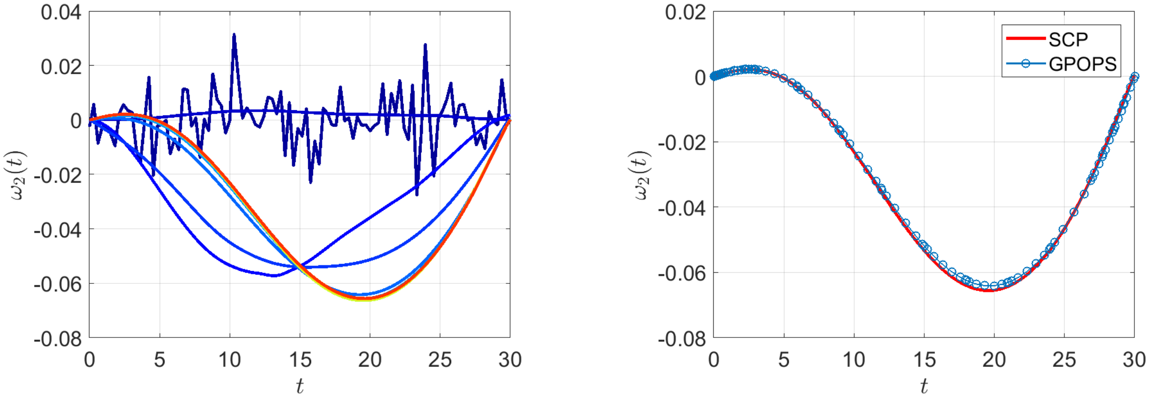
Fig. 15 Case 2: Convergence and Comparison of q_4



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

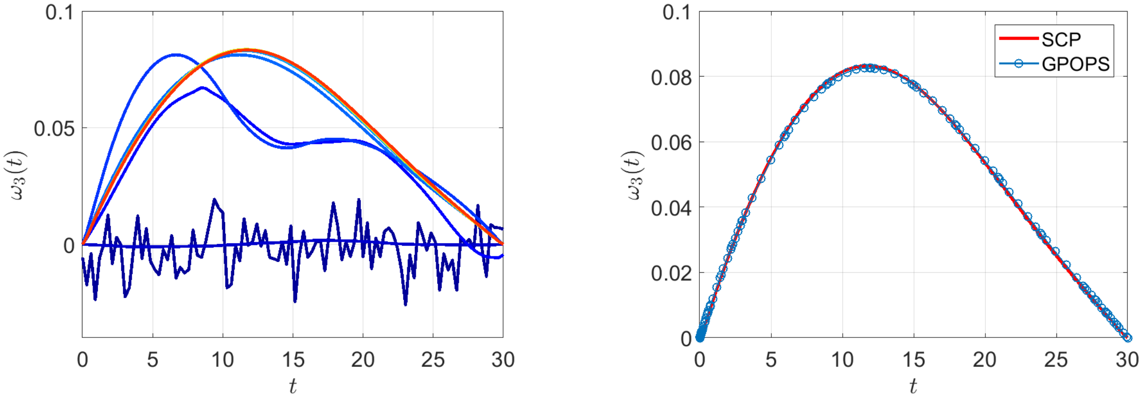
Fig. 16 Case 2: Convergence and Comparison of ω_1



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

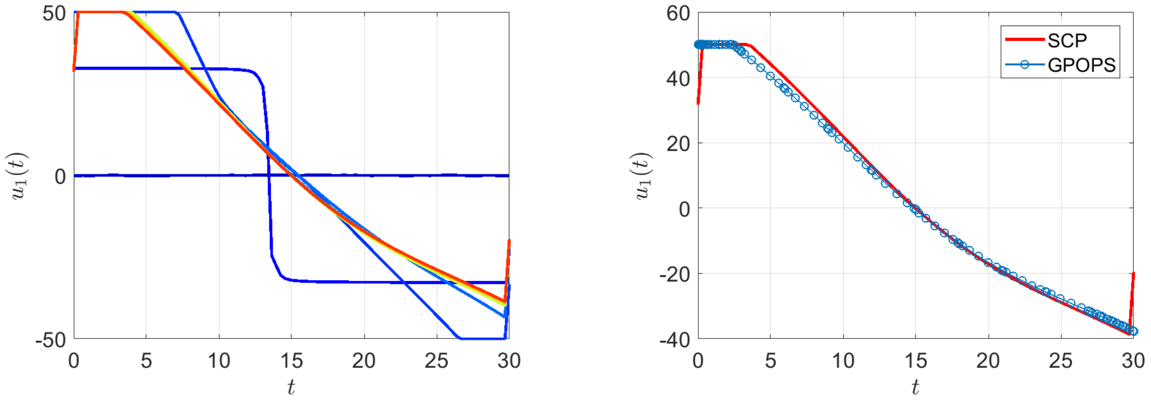
Fig. 17 Case 2: Convergence and Comparison of ω_2



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

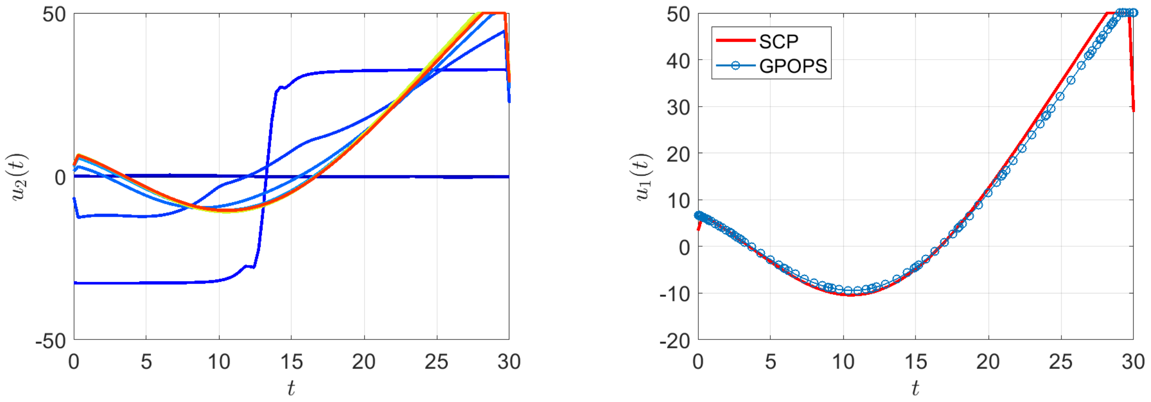
Fig. 18 Case 2: Convergence and Comparison of ω_3



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

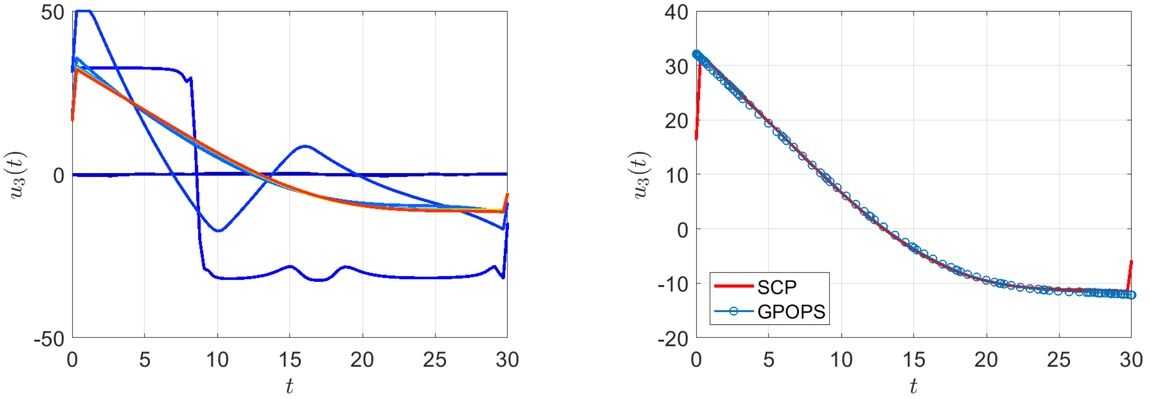
Fig. 19 Case 2: Convergence and Comparison of u_1



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

Fig. 20 Case 2: Convergence and Comparison of u_2



(a) LS-SCP Convergence

(b) LS-SCP Comparison with GPOPS

Fig. 21 Case 2: Convergence and Comparison of u_3

V. Conclusion

In this paper, a sequential convex programming method is designed to solve the trajectory optimization problem for the minimum-energy fixed-time reorientation of an asymmetric rigid body spacecraft. When presented in its original form, the highly nonlinear dynamics make analytical solutions challenging to impossible to solve in general. While NLP solvers may provide solutions, the problems of convergence and computational load make them hard to implement in practice. However, in this paper, the original problem is translated into a sequence of convex QP problems. This is accomplished through Jacobian approximation and affine equality constraints to enforce dynamics. Additionally, a simple line search routine is employed to aid convergence of the SCP method. The effectiveness of the combined LS-SCP method is then demonstrated through numerical demonstrations using parameters found from the NASA XTE spacecraft. Two cases are considered. The first of which is a small angle slew, of which, the LS-SCP method was shown to converge to the optimal solution of the original problem and verified using the NLP solver GPOPS. In case 1, it is also shown that the addition of LS aids convergence significantly by reducing the number of SCP iterations from 11 to 7. In case 2, a large angle slew is considered and the LS-SCP method is shown to converge to the optimal solution even when using random values as the initial trajectory to SCP. With this said, it is expected the LS-SCP method presented in this paper provides a rapid, reliable, and robust solution to original minimum-energy fixed-time reorientation of an asymmetric spacecraft.

References

[1] Fleming, A., Sekhavat, P., and Ross, I. M., "Minimum-Time Reorientation of a Rigid Body," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 1, 2010.

[2] Yang, C. C., and J, W. C., "Optimal Large-Angle Attitude Control of Rigid Spacecraft by Momentum Transfer," *IET Control Theory Applications*, Vol. 1, No. 3, 2007.

[3] Marsh, H. C., Karpenko, M., and Gong, Q., "Relationships Between Maneuver Time and Energy for Reaction Wheel Attitude Control," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 2, 2018.

[4] Liu, S., and Singh, T., "Fuel/Time Optimal Control of Spacecraft Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 2, 2012.

[5] Ioslovich, I., "Arbitrary Fuel-Optimal Attitude Maneuvering of a Non-Symmetric Space Vehicle in a Vehicle-Fixed Coordinate Frame," *Automatica*, Vol. 39, No. 3, 2003, pp. 557–562.

[6] Dixon, M. V., Edelmaum, T. N., Potter, J. E., and Vandervelde, W. E., "Fuel Optimal Reorientation of Axisymmetric Spacecraft," *Journal of Spacecraft and Rockets*, Vol. 7, No. 11, 1970.

[7] Seywald, H., Kumar, R. R., Deshpande, S. S., and Heck, M. L., "Minimum Fuel Spacecraft Reorientation," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 1, 1994.

[8] Zhuang, Y., Ma, G., Huang, H., and Li, C., "Real-time Trajectory Optimization of an Underactuated Rigid Spacecraft Using Differential Flatness," *Aerospace Science and Technology*, Vol. 23, No. 1, 2012.

[9] Boyarko, G., Romano, M., and Yakimenko, O. A., "Time-Optimal Reorientation of a Spacecraft Using an Inverse Dynamics Optimization Method," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, 2011.

[10] Acikmese, B., and Ploen, S. R., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366.

[11] Blackmore, L., Acikmese, B., and Scharf, D. P., "Minimum Landing Error Powered Descent Guidance for Mars Landing Using Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, 2010, pp. 1161–1171.

[12] Liu, X., Shen, Z., and Lu, P., "Solving the Maximum-Crossrange Problem via Successive Second-Order Cone Programming with a Line Search," *Aerospace Science and Technology*, Vol. 47, No. 1, 2015.

[13] Liu, X., Shen, Z., and Lu, P., "Exact Convex Relaxation for Optimal Flight of Aerodynamically Controlled Missiles," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 52, No. 4, 2016, pp. 1881–1892.

[14] Liu, X., Lu, P., and Pan, B., "Survey of Convex Optimization for Aerospace Applications," *Astrodynamics*, Vol. 1, No. 1, 2017, pp. 23–40.

[15] Wang, Z., and Grant, M. J., "Optimization of Minimum-Time Low-Thrust Transfers Using Convex Programming," *Journal of Spacecraft and Rockets*, Vol. 55, No. 3, 2017, pp. 586–598.

[16] Wang, Z., and Grant, M. J., "Constrained Trajectory Optimization for Planetary Entry via Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 10, 2017, pp. 2603–2615.

[17] Wang, Z., and Grant, M. J., "Minimum-Fuel Low-Thrust Transfers for Spacecraft: A Convex Approach," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, No. 5, 2018, pp. 2274–2290.

[18] Wang, Z., and Grant, M. J., "Autonomous Entry Guidance for Hypersonic Vehicles by Convex Optimization," *Journal of Spacecraft and Rockets*, Vol. 55, No. 4, 2018, pp. 993–1006.

[19] McDonald, S. T., and Wang, Z., "Real-Time Optimal Trajectory Generation for UAV to Rendezvous with an Aerial Orbit," *AIAA Aviation 2019 Forum*, Dallas, Texas, 2019.

[20] Wang, Z., and McDonald, S. T., "Optimal Rendezvous of Unmanned Aerial and Ground Vehicles via Sequential Convex Programming," *AIAA Aviation 2019 Forum*, Dallas, Texas, 2019.

[21] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, Cambridge, England, 2004, pp. 127–189.

[22] Wright, S. J., *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, PA, 1997, pp. 1–47.

[23] Kim, Y., Mesbahi, M., and Hadaegh, G. S. F. Y., "On the Convex Parameterization of Constrained Spacecraft Reorientation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 46, No. 3, 2010.

[24] Sun, C., and Dai, R., "Spacecraft Attitude Control under Constrained Zones via Quadratically Constrained Quadratic Programming," *AIAA Guidance, Navigation, and Control Conference*, Kissimmee, Florida, 2015.

[25] Wang, Z., "Optimal Trajectories and Normal Load Analysis of Hypersonic Glide Vehicles via Convex Optimization," *Aerospace Science and Technology*, Vol. 87, 2019, pp. 357–368.

[26] Domahidi, A., Chu, E., and Boyd, S., "ECOS: an SOCP Solver for Embedded System," *European Control Conference*, Zurich, Switzerland, 2013, pp. 3071–3076.