JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 42, No. 8, August 2019

Experimental Validation of Pseudospectral-Based Optimal Trajectory Planning for Free-Floating Robots

Alexander Crain* and Steve Ulrich†
Carleton University, Ottawa, Ontario K1S 5B6, Canada

DOI: 10.2514/1.G003528

This paper proposes the use of pseudospectral methods to solve the nonlinear trajectory planning problem for free-floating robots. Specifically, three different optimization tools are analyzed. Using each tool, simulations are performed, and it is shown that each solver is capable of finding a deployment trajectory that minimizes the final attitude of a free-floating robot. Each solution is then validated using Pontryagin's minimum principle and Bellman's principle of optimality, as well as by propagating the control torques using a numerical integrator and the dynamics model. Experimental validation is performed at Carleton University's Spacecraft Robotics and Control Laboratory to further investigate the solutions obtained from each tool. Ultimately, it was determined that all solutions resulted in a reduced attitude disturbance at the end of the robotic deployment maneuver.

I. Introduction

THERE are currently over 500,000 pieces of debris being tracked as they orbit around the Earth. Sources of large debris include malfunctioning or decommissioned spacecraft and depleted rocket engines, most of which are travelling at speeds of up to 28,000 km/h. It is now widely known that, even with no future launches, orbital debris has reached the point where any collisions among large-body debris will lead to an unstable growth in debris [1]. This was predicted over 30 years ago, when the term "Kessler Syndrome" was coined. Kessler Syndrome, in brief, refers to the concept of collisional cascading of objects. Two orbiting objects that pass through the same distance from the object that they are orbiting about will eventually collide [2,3] and break up into a number of smaller fragments, thus creating an even larger number of objects. However, research has also shown that removing as few as five large objects each year can stabilize debris growth [2].

One of the main technological challenges inherent to such missions occurs during the autonomous robotic capture of uncooperative targets. As with any general free-floating object, the nonlinear coupling between the base and the robotic manipulator may induce an undesirable deviation in the attitude of the base, which can in turn cause the destabilization of the free-floating robot (FFR) or severe damage to the robotic arm if considerations are not made for this deviation. The most common approach to solving this issue is the concept of trajectory optimization, which involves determining the best set of control inputs to minimize a given cost function.

The list of possible maneuvers that could benefit from trajectory optimization is extensive, yet there have been many researchers who have proposed solutions to the problem of optimal trajectory planning in the context of robotics. Dubowsky and Torres [4] and Torres and Dubowsky [5] worked on preliminary path planning for space manipulators by presenting a technique called the "enhanced disturbance map." This map was used to aid in the understanding of the complex problem of trajectory optimization, as well as in the development of algorithms to reduce disturbances. Agrawal and Xu [6] proposed a global optimum path planning technique for redundant space manipulators. This technique used a variational approach to minimize the objective function; that is to say, the holonomic and nonholonomic constraint equations for the free-floating robot were adjoined to the objective function using the Lagrange multiplier technique, which led to a system of differential and algebraic equations that were then numerically integrated. Papadopoulos and Abu-Abed [7] designed a three-degree-of-freedom (3-DOF) redundant manipulator based on a force and moment transmission analysis, with the objective of achieving reactionless motions. This was achieved by eliminating the dynamic reaction forces through force balancing, whereas reaction moments were eliminated by following reactionless paths. However, to simplify the problem, the authors chose to render the dynamics of the system decoupled and invariant.

Lampariello et al. [8] presented a time-optimal motion planning method for free-flying robots while considering full-state actuation such that large spacecraft displacements were allowed. The proposed solution from the authors did not require any spacecraft actuation to achieve the end goal because the robot motion alone proved to be sufficient for the proposed task. Aghili [9] focused on the guidance of a robot manipulator to capture and detumble a spacecraft with unknown dynamic parameters. A Kalman filter was used to estimate the states and all dynamic parameters, and then the optimal control actions were determined such that the time of travel and the weighted norms of the end-effector (EE) velocity and acceleration were minimized. Experimental validation of the proposed technique was also performed, and successful grasping was achieved despite the unknown parameters of the target spacecraft. Yoshida et al. [10] discussed the minimization of the base attitude deviation before and after the contact with the target from the viewpoint of angular momentum distribution. Their proposed methodology for the capture and detumbling of a spacecraft used a bias momentum approach during the approaching phase, impedance control during the impact, and distributed momentum control during the postimpact phase.

Huang and Xu [11] proposed a particle swarm optimization strategy to determine globally time-optimal trajectories for free-floating robots, and the control strategy demonstrated satisfactory performance. Oki et al. [12] investigated an optimal control method for capturing a tumbling spacecraft, although they focused primarily on minimizing the operational time for fast capturing while still paying attention to limitations on the grasping force and torque, detumbling operations along an arbitrary arm motion, and parameter uncertainty in the target spacecraft. This concept of detumbling speed proved to be beneficial when handling targets with uncertain dynamics. Cascio [13], meanwhile, looked at the concept of optimal path planning for multiarm, multilink fixed-base robotic manipulators using pseudospectral (PS) methods. The optimal control problem was posed for fixed-based manipulators only, yet the resulting optimization was still highly nonlinear and was solved using the software package DIDO. The optimal solution obtained from the software package was verified using Pontryagin's minimum principle.

Received 3 January 2018; revision received 8 January 2019; accepted for publication 24 January 2019; published online 8 March 2019.

*Graduate Student, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Student Member AIAA.

†Associate Professor, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Senior Member AIAA.

Liu et al. [14] proposed an optimal trajectory planning method for the stabilization of coupled space robots following a capture using particle swarm optimization. In their work, the trajectory planning problem was transformed into a nonlinear optimization problem in which the attitude deflection of the base was used as the cost function. The authors found that the particle swarm optimization technique was effective at solving the problem. Xu et al. [15] investigated an adaptive version of reaction null-space-based control for free-floating robots with uncertain kinematics and dynamics. The contribution of their work lies in the derivation of a linear expression for use as the basis of the adaptive control; the authors ultimately found that their proposed controller achieved both the base attitude regulation and the continuous path tracking of the EE. Flores-Abad et al. [16] showed that disturbances to the attitude of the free-floating robot could be greatly reduced through offline nonlinear optimal guidance laws that made use of PS techniques to predict the optimal manipulator trajectory. Similar to the work published by Cascio [13], Flores-Abad et al. [16] solved the nonlinear optimization problem using the software package TOMLAB. These results were independently validated by Crain and Ulrich [17], in which the nonlinear optimization problem was solved using the software package GPOPS-I, and it was determined that both tools converged to approximately the same local minimum.

Rybus et al. [18] presented an improved manipulator trajectory optimization method for systems with nonconservative momentum. Their proposed technique allowed for the minimization of a quadratic norm connected with the power use of motors in manipulator joints. This differs from existing techniques due to the fact that it includes the possibility to constrain the final velocity of the EE by making the period of the capture maneuver an optimization variable. Virgili-Llop et al. [19,20] described a set of laboratory-based experiments using a free-floating robot in which they demonstrated the autonomous capture of a nonmoving space object using a robotic manipulator. The authors made use of the Proximity Operation of Spacecraft: Experimental hardware-In-the-loop DYNamic Simulator testing platform [21], which can create a quasi-frictionless planar environment. Using this testing platform, the authors were able to implement and test a variety of control methods in a dynamically accurate environment.

Other publications in the field includes the work of Misra and Bai [22], which consisted of task-constrained trajectory planning for a free-floating robot system using convex optimization. In brief, based on the conservation of the system moment, Misra and Bai [22] formulated a trajectory planner as a convex quadratic programming problem in joint space. By formulating the problem in such a way, they were able to rapidly generate optimal trajectories for the EE, which in turn minimized the disturbance to the base of the space robot. Additionally, Misra and Bai [23] have published work in the field of optimal path planning for free-flying space robots, in which the attitude of the base spacecraft was actively controlled. The main contribution was in the formulation of the final EE pose constraint and its subsequent convex relaxation; specifically, the proposed formulation allowed the problem to be solved efficiently through convex optimization. Finally, Yoshida et al. [24] presented experimental results for a zero-reaction maneuver, which used reaction null-space-based reactionless manipulation to successfully deploy the manipulator of the Engineering Test Satellite No. 7 while ensuring zero-attitude displacement for the base.

With the advent of smaller yet more powerful computers, the usefulness of offline trajectory optimization using computationally heavy techniques such as pseudospectral optimization methods has become more viable than ever before. There exists a plethora of software tools that are designed to hide the bulk of the background calculations from the user, thus creating turnkey software that requires minimal effort from the user. However, the problem of optimal control is more than the implementation of the problem, and each software tool has different nuances, which can be confusing to first time users.

In this work, three different pseudospectral optimization tools are used to determine optimal trajectories for a FFR with a 3-DOF robotic manipulator. The three tools, GPOPS-I, DIDO, and TOMLAB, will herein be referred to as "OPTIM-TOOL 1," "OPTIM-TOOL 2," and "OPTIM-TOOL 3," respectively. The nonlinear problem to be addressed takes into consideration the constraints on the system while minimizing the final attitude. Contrary to past work in this area [16,17], the minimization of any running costs is ignored in favor of simply minimizing the endpoint of the attitude. By minimizing the endpoint, the manipulator is less constrained in terms of potential trajectories, thus reducing the overall computational effort required to converge to a solution. However, as outlined in the body of this paper, focusing solely on the minimization of the endpoint does come at the cost of a larger transient attitude displacement as well as an increase in the energy usage for the joints. Despite these disadvantages, which may be undesirable depending on the application, the focus of this optimal guidance problem is on demonstrating the advantage of manipulating conservation of the angular momentum to ensure a net zero-attitude disturbance deployment. Additionally, this proposed technique also reduces both fuel consumption and the saturation level of the reaction wheels because it will no longer be necessary to correct the attitude postdeployment.

Furthermore, each of the software tools used in this paper is compared to determine the viability of each one in their default settings. The results from each optimal solution are verified using three important steps: first, the solutions are numerically checked using Pontryagin's minimum principle; then, Bellman's principle of optimality is used to verify that an optimal control policy has been found; and finally, the optimal solutions from each tool are propagated forward using a numerical integrator to determine whether or not the solutions followed the dynamic constraints. Following the verification of the results obtained, the optimal solutions are experimentally validated at Carleton University's Spacecraft Robotics and Control Laboratory (SRCL), using the Spacecraft Proximity Operations Testbed (SPOT); for each experiment, a repeatability analysis is performed. This paper has several significant contributions: the first contribution lies in the formulation and subsequent solving of the nonlinear deployment problem for the FFR using PS techniques. Furthermore, contrary to most past work, this paper also validates the solution by solving the problem using different tools and comparing the results from each tool. This paper also provides the steps for validating an optimal solution in a clear and concise manner, and it shows how these criteria confirm the results obtained are correct, which is an important step often overlooked in this area of research because most publications assume these tools to be black boxes. Lastly, the experimental validation of the optimal solutions presents a significant contribution because it is highly uncommon in the field of trajectory planning for FFRs [16,25,26].

This paper is organized as follows: Sec. II outlines the general kinematic and dynamic equations that describe the manipulator motion. Section III details the optimal trajectory planning technique used, as well as the associated equations. Section IV presents simulation results for each optimal control tool, whereas Sec. V verifies the optimality of the solutions obtained from each tool. Section VI provides background information on the experimental testbed, summarizes the experimental procedure, and introduces the experimental results with a brief repeatability analysis; several sources of error are also listed. Concluding remarks are then made in Sec. VII.

II. Kinematics and Dynamics of Space Robots

In this section, the kinematics and dynamics for the general case of a space robot are summarized. The section will also include the case-specific dynamics for the planar space robot used in SPOT, which is shown in Sec. VI.

A. Primary Assumptions

Determining the kinematics and dynamics model for a space robot is a problem that has been addressed several times in the past, but it will still be described in detail here for completeness. First, the derivation of the dynamics and kinematics model chosen for this paper requires the following assumptions:

1) The spacecraft is assumed to be a rigid body, and joint friction is neglected. These assumptions allow for a simpler dynamics model.

2) Because the primary focus of this paper is the experimental validation of the technique using the laboratory at Carleton University, no external perturbations typically found on orbit are included in the model.

3) The kinematics and dynamics are exclusively planar in nature. No considerations are made in the model development for spatial motion.

4) The total momentum of the system is assumed to be conserved; this assumption should be valid in a frictionless environment for the short duration of the deployment maneuver.

B. Kinematics Model

The forward kinematics for a space robot involves determining the EE position in inertial space. This can be geometrically written as follows:

$$\boldsymbol{p}_e = \boldsymbol{p}_0 + \boldsymbol{b}_0 + \sum_{i=1}^{n} \boldsymbol{L}_i \tag{1}$$

where $\boldsymbol{p}_e \in \mathbb{R}^3$ is the position vector pointing from the origin of the inertial reference frame to the EE, $\boldsymbol{p}_0 \in \mathbb{R}^3$ is the position vector pointing from the origin of the inertial reference frame to the center of mass (CM) of the spacecraft body, $\boldsymbol{b}_0 \in \mathbb{R}^3$ is the position vector pointing from the CM to the first joint i, $\boldsymbol{L}_i \in \mathbb{R}^3$ is the position vector pointing from joint i to joint $i+1$, and n is the number of links forming the manipulator.

C. Dynamics Model

The nonlinear dynamic equation of a multilink robot with rigid links is derived in terms of kinetic and potential energies stored in the system by the Euler–Lagrange formulation as outlined in Ref. [27]. Ultimately, the dynamics for any n-link serial manipulator can be condensed into the more convenient form shown as follows:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\tau} \tag{2}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is referred to as the inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the centrifugal/Coriolis matrix, and $\boldsymbol{\tau}$ is the vector of generalized forces and torques. From here, Eq. (2) can then be partitioned into a more useful form as detailed in Ref. [24]:

$$\begin{bmatrix} \boldsymbol{M}_b & \boldsymbol{M}_{bm} \\ \boldsymbol{M}_{bm}^T & \boldsymbol{M}_m \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_b \\ \ddot{\boldsymbol{\theta}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_b \\ \boldsymbol{\tau}_c \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_b^T \\ \boldsymbol{J}_m^T \end{bmatrix} \boldsymbol{f}_e \tag{3}$$

where $\boldsymbol{M}_b \in \mathbb{R}^{6\times6}$ is the inertia matrix for the base; $\boldsymbol{M}_m \in \mathbb{R}^{n\times n}$ is the inertia matrix for the links; $\boldsymbol{M}_{bm} \in \mathbb{R}^{6\times n}$ is the coupled inertia matrix; $\ddot{\boldsymbol{x}}_b \in \mathbb{R}^6$ is the linear and angular acceleration vector for the base spacecraft; $\ddot{\boldsymbol{\theta}} \in \mathbb{R}^n$ is the manipulator joint angular acceleration vector; $\boldsymbol{c}_b \in \mathbb{R}^6$ is the velocity dependent nonlinear matrix for the base; $\boldsymbol{c}_m \in \mathbb{R}^n$ is the velocity dependent nonlinear matrix for the manipulator; $\boldsymbol{f}_b \in \mathbb{R}^6$ is the vector of forces and moments exerted on the CM of the base; $\boldsymbol{f}_e \in \mathbb{R}^6$ is the vector of forces and moments exerted on the EE; $\boldsymbol{\tau}_c \in \mathbb{R}^n$ is the vector of joint torques; $\boldsymbol{J}_b \in \mathbb{R}^{6\times6}$ is the Jacobian matrix for the base; and, finally, $\boldsymbol{J}_m \in \mathbb{R}^{6\times n}$ is the Jacobian matrix for the manipulator.

D. Generalized Jacobian

It is important to review the concept of the generalized Jacobian here because the transpose Jacobian control law used in this paper requires the analytical expression for the generalized Jacobian. The concept of the generalized Jacobian was first introduced by Umetani and Yoshida [28]. The generalized Jacobian can be expressed in terms of the partitions presented in Eq. (3):

$$\boldsymbol{J}^* = \boldsymbol{J}_m - \boldsymbol{J}_b \boldsymbol{M}_b^{-1} \boldsymbol{M}_{bm} \tag{4}$$

where $\boldsymbol{J}^* \in \mathbb{R}^{6\times n}$ is the generalized Jacobian matrix. Using this matrix, a simple control law for a space robot can be developed [29,30]; this is typically referred to as a transpose Jacobian control law:

$$\boldsymbol{\tau}_c = (\boldsymbol{J}^*)^T \left(\boldsymbol{K}_p \boldsymbol{x}_e + \boldsymbol{K}_d \frac{\mathrm{d}\boldsymbol{x}_e}{\mathrm{d}t} + \boldsymbol{K}_I \int \boldsymbol{x}_e \, \mathrm{d}t \right) \tag{5}$$

where $\boldsymbol{x}_e \in \mathbb{R}^6$ is the error between the reference and the actual EE pose, $\boldsymbol{K}_p \in \mathbb{R}^{6\times6}$ is the proportional gain matrix, $\boldsymbol{K}_d \in \mathbb{R}^{6\times6}$ is the derivative gain matrix, and $\boldsymbol{K}_I \in \mathbb{R}^{6\times6}$ is the integral gain matrix. The dimensions listed for the gains in this controller assume the manipulator has six degrees of freedom: the dimensions will differ for different types of joints.

E. Kinematics and Dynamics of the Spacecraft Proximity Operations Testbed Space Robot

In this section, the forward kinematics, inverse kinematics, and dynamics for the experimental platform used at Carleton University are derived. The spacecraft used in the experimental validation campaign is shown in Sec. VI, and a representation of this spacecraft (including the definitions for the reference frames used and the generalized states) is shown in Fig. 1. Referring to the geometry of the planar space robot shown in Fig. 1, the kinematics at position level for this system can be derived using Eq. (1). The resulting equations are given by the following:

$$\begin{aligned} x_{\mathrm{ee}} = {} & x_0 + b_0 \cos(\phi + q_0) + (a_1 + b_1)\cos(\pi/2 + q_0 + q_1) \\ & + (a_2 + b_2)\cos(\pi/2 + q_0 + q_1 + q_2) \\ & + (a_3 + b_3)\cos(\pi/2 + q_0 + q_1 + q_2 + q_3) \end{aligned} \tag{6}$$

$$\begin{aligned} y_{\mathrm{ee}} = {} & y_0 + b_0 \sin(\phi + q_0) + (a_1 + b_1)\sin(\pi/2 + q_0 + q_1) \\ & + (a_2 + b_2)\sin(\pi/2 + q_0 + q_1 + q_2) \\ & + (a_3 + b_3)\sin(\pi/2 + q_0 + q_1 + q_2 + q_3) \end{aligned} \tag{7}$$

where $x_{\mathrm{ee}} \in \mathbb{R}$ and $y_{\mathrm{ee}} \in \mathbb{R}$ are the scalar components of the EE position in the inertial frame, $x_0 \in \mathbb{R}$ and $y_0 \in \mathbb{R}$ are the positions of spacecraft CM in the inertial frame, $q_i \in \mathbb{R}$ is the principle angle of rotation for each body i, $b_0 \in \mathbb{R} \geq 0$ is the magnitude of the vector $\boldsymbol{b}_0$, $a_i \in \mathbb{R} \geq 0$ is the distance from joint i to the CM of link i, $b_i \in \mathbb{R} \geq 0$ is the distance from the CM of link i and joint $i+1$, and $\phi \in \mathbb{R}$ is the angle describing the location of the first joint in the body-fixed reference frame. The scalar components of the EE velocity, $\dot{x}_{\mathrm{ee}} \in \mathbb{R}$ and $\dot{y}_{\mathrm{ee}} \in \mathbb{R}$, can be obtained by differentiating Eqs. (6) and (7) with respect to time.

Classically, the inverse kinematics for a space robot is concerned with determining the states of the system knowing the inertial position of the EE. In this paper, inverse kinematics are applied to

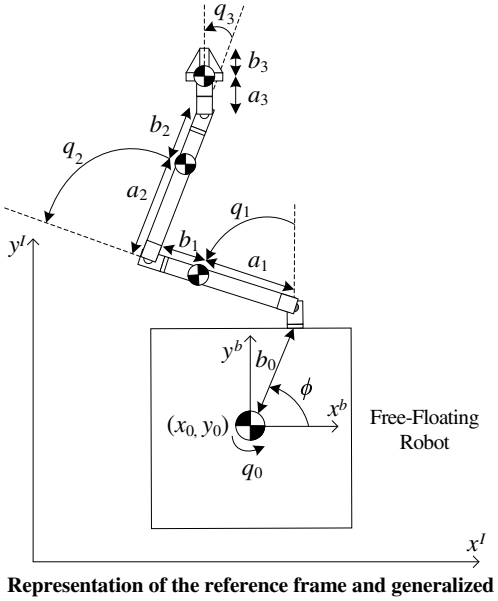


Fig. 1 Representation of the reference frame and generalized states.

determine the manipulator joint angles during any given experiment; the inertial location and attitude of the spacecraft, as well as the EE position, wrist position, and elbow position, are all measured quantities during any experiment. Thus, the shoulder joint angle can be obtained using the following:

$$q_1 = \operatorname{atan2}\Big(y_{\text{el}} - y_0 - b_0 \sin(\phi + q_0), x_{\text{el}} - x_0 - b_0 \cos(\phi + q_0)\Big) - \frac{\pi}{2} - q_0 \tag{8}$$

Then, the elbow and wrist joints can be obtained from the following:

$$q_2 = \operatorname{atan2}(y_w - y_{\text{el}}, x_w - x_{\text{el}}) - \frac{\pi}{2} - q_0 - q_1 \tag{9}$$

and

$$q_3 = \operatorname{atan2}(y_{\text{ee}} - y_w, x_{\text{ee}} - x_w) - \frac{\pi}{2} - q_0 - q_1 - q_2 \tag{10}$$

where $x_w \in \mathbb{R}$ and $y_w \in \mathbb{R}$ are the scalar components of the wrist position in the inertial frame, and $x_{\text{el}} \in \mathbb{R}$ and $y_{\text{el}} \in \mathbb{R}$ are the scalar components of the elbow position in the inertial frame. For any n-link manipulator, the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ can be derived in terms of the linear and angular velocity components of the kinetic energy of the system, which are functions of the Jacobian matrix and the derivatives of the joint variables. The Jacobian matrix is a more convenient way to formulate the velocity kinematics of the manipulator. A suitable Jacobian that relates the linear and angular velocity of the CM for each body, denoted by $\boldsymbol{v}_c, \boldsymbol{\omega}_c \in \mathbb{R}^3$, to the joint velocity, can be obtained as follows:

$$\boldsymbol{v}_c = \boldsymbol{J}_v(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{11}$$

$$\boldsymbol{\omega}_c = \boldsymbol{J}_\omega(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{12}$$

where by $\boldsymbol{J}_v(\boldsymbol{q}), \boldsymbol{J}_\omega(\boldsymbol{q}) \in \mathbb{R}^{3\times 6}$ and Eqs. (11) and (12) can be grouped together and represented by the following:

$$[\,v_{\text{cx}} \quad v_{\text{cy}} \quad v_{\text{cz}} \quad \omega_{\text{cx}} \quad \omega_{\text{cy}} \quad \omega_{\text{cz}}\,]^T = \boldsymbol{J}_c[\,\dot{x}_0 \quad \dot{y}_0 \quad \dot{q}_0 \quad \dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3\,]^T \tag{13}$$

where

$$\boldsymbol{J}_c = \begin{bmatrix} \boldsymbol{J}_v(\boldsymbol{q}) \\ \boldsymbol{J}_\omega(\boldsymbol{q}) \end{bmatrix} \tag{14}$$

Then, the Jacobian for each body is given by the following:

$$\boldsymbol{J}_{c_0}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_{v_0}(\boldsymbol{q}) \\ \boldsymbol{J}_{\omega_0}(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{15}$$

Similarly, this can be done for the first joint as follows:

$$\boldsymbol{J}_{c_1}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_{v_1}(\boldsymbol{q}) \\ \boldsymbol{J}_{\omega_1}(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & J_{c_1}[1,3] & J_{c_1}[1,4] & 0 & 0 \\ 0 & 1 & J_{c_1}[2,3] & J_{c_1}[2,4] & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{16}$$

Then, for the second joint, the Jacobian is given by the following:

$$\boldsymbol{J}_{c_2}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_{v_2}(\boldsymbol{q}) \\ \boldsymbol{J}_{\omega_2}(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & J_{c_2}[1,3] & J_{c_2}[1,4] & J_{c_2}[1,5] & 0 \\ 0 & 1 & J_{c_2}[2,3] & J_{c_2}[2,4] & J_{c_2}[2,5] & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{17}$$

Finally, for the third joint, the Jacobian is as follows:

$$\boldsymbol{J}_{c_3}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_{v_3}(\boldsymbol{q}) \\ \boldsymbol{J}_{\omega_3}(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & J_{c_3}[1,3] & J_{c_3}[1,4] & J_{c_3}[1,5] & J_{c_3}[1,6] \\ 0 & 1 & J_{c_3}[2,3] & J_{c_3}[2,4] & J_{c_3}[2,5] & J_{c_3}[2,6] \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{18}$$

where

$$\begin{aligned}
J_{c_1}[1,3] &= -b_0 S_0 - a_1 S_1 \\
J_{c_1}[1,4] &= -a_1 S_1 \\
J_{c_1}[2,3] &= b_0 C_0 + a_1 C_1 \\
J_{c_1}[2,4] &= a_1 C_1 \\
J_{c_2}[1,3] &= -b_0 S_0 - L_1 S_1 - a_2 S_2 \\
J_{c_2}[1,4] &= -L_1 S_1 - a_2 S_2 \\
J_{c_2}[1,5] &= -a_2 S_2 \\
J_{c_2}[2,3] &= b_0 C_0 + L_1 C_1 + a_2 C_2 \\
J_{c_2}[2,4] &= L_1 C_1 + a_2 C_2 \\
J_{c_2}[2,5] &= a_2 C_2 \\
J_{c_3}[1,3] &= -b_0 S_0 - L_1 S_1 - L_2 S_2 - a_3 S_3 \\
J_{c_3}[1,4] &= -L_1 S_1 - L_2 S_2 - a_3 S_3 \\
J_{c_3}[1,5] &= -L_2 S_2 - a_3 S_3 \\
J_{c_3}[1,6] &= -a_3 S_3 \\
J_{c_3}[2,3] &= b_0 C_0 + L_1 C_1 + L_2 C_2 + a_3 C_3 \\
J_{c_3}[2,4] &= L_1 C_1 + L_2 C_2 + a_3 C_3 \\
J_{c_3}[2,5] &= L_2 C_2 + a_3 C_3 \\
J_{c_3}[2,6] &= a_3 C_3
\end{aligned}$$

and

$$\begin{aligned}
S_0 &= \sin(\phi + q_0) \\
S_1 &= \sin(\pi/2 + q_0 + q_1) \\
S_2 &= \sin(\pi/2 + q_0 + q_1 + q_2) \\
S_3 &= \sin(\pi/2 + q_0 + q_1 + q_2 + q_3) \\
C_0 &= \cos(\phi + q_0) \\
C_1 &= \cos(\pi/2 + q_0 + q_1) \\
C_2 &= \cos(\pi/2 + q_0 + q_1 + q_2) \\
C_3 &= \cos(\pi/2 + q_0 + q_1 + q_2 + q_3)
\end{aligned}$$

Hence, the translational contribution to the total kinetic energy, $\boldsymbol{T}_v \in \mathbb{R}^{6\times 6}$, is as follows:

$$\boldsymbol{T}_v = \frac{1}{2}\dot{\boldsymbol{q}}^T\left(m_0\boldsymbol{J}_{v_0}^T\boldsymbol{J}_{v_0} + m_1\boldsymbol{J}_{v_1}^T\boldsymbol{J}_{v_1} + m_2\boldsymbol{J}_{v_2}^T\boldsymbol{J}_{v_2} + m_3\boldsymbol{J}_{v_3}^T\boldsymbol{J}_{v_3}\right)\dot{\boldsymbol{q}} \tag{19}$$

Similarly, the rotational contribution to the total kinetic energy, $\boldsymbol{T}_\omega \in \mathbb{R}^{6\times 6}$, is as follows:

$$\boldsymbol{T}_\omega = \frac{1}{2}\dot{\boldsymbol{q}}^T\left(I_0\boldsymbol{J}_{\omega_0}^T\boldsymbol{J}_{\omega_0} + I_1\boldsymbol{J}_{\omega_1}^T\boldsymbol{J}_{\omega_1} + I_2\boldsymbol{J}_{\omega_2}^T\boldsymbol{J}_{\omega_2} + I_3\boldsymbol{J}_{\omega_3}^T\boldsymbol{J}_{\omega_3}\right)\dot{\boldsymbol{q}} \tag{20}$$

Therefore, the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ for the case presented in this paper is given by the sum of the linear and angular kinetic energy, and it results in the following matrix:

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} & M_{26} \\ M_{31} & M_{32} & M_{33} & M_{34} & M_{35} & M_{36} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} & M_{46} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} & M_{56} \\ M_{61} & M_{62} & M_{63} & M_{64} & M_{65} & M_{66} \end{bmatrix} \tag{21}$$

where $M_{ij}\forall i = 1\ldots 6,\ j = 1\ldots 6$ are defined in the Appendix. Using the components of the inertia matrix, the Christoffel symbols can be derived and the centrifugal/Coriolis matrix can be obtained:

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \tag{22}$$

where $C_{ij}\forall i = 1\ldots 6,\ j = 1\ldots 6$ are also defined in the Appendix.

III. Optimal Trajectory Planning: Approach Overview

In this paper, the optimal trajectory problem was defined and solved within the MATLAB-Simulink environment using three different commercial optimization tools. In each case, the sparse nonlinear optimizer (SNOPT) was used to solve the optimal control problem. For completeness, a summary of the technique used in OPTIM-TOOLs 1 and 3 is presented in the following section; the technique used in OPTIM-TOOL 2 is similar, but it will not be covered. For a more comprehensive description of each tool, interested readers may refer to the following: for OPTIM-TOOL 1, refer to the works of Benson [31], Huntington [32], Benson et al. [33], Huntington et al. [34], and Huntington and Rao [35]; for further information on OPTIM-TOOL 2, refer to the works of Ross and Karpenko [36,37]; finally, for OPTIM-TOOL 3, readers may refer to the User's Guide for TOMLAB/SOL [38], as well as the work of Gill et al. [39].

A. Gauss Pseudospectral Methods

Each optimization tool was implemented alongside the dynamics model presented in Sec. II to obtain the results, as well as to validate the capability of the MATLAB-Simulink environment to accurately optimize the manipulator trajectory. Each tool requires the user to define the dynamics of the problem using differential equations, an associated cost function, and finally the limits and initial guesses for each of the states and controls.

In its general form, any optimal guidance/control problem can be formulated so as to minimize the cost function:

$$J = \boldsymbol{\Phi}(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\,\mathrm{d}t \tag{23}$$

where $\boldsymbol{\Phi}$ is the Mayer component, and $\boldsymbol{g}$ is the Lagrange component. The system is subject to the dynamic constraints given by the following:

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \tag{24}$$

and the associated boundary conditions:

$$\boldsymbol{\phi}\left(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f\right) = 0 \tag{25}$$

For this research, the optimal guidance problem defined in Eqs. (23–25) is solved by OPTIM-TOOLs 1 and 3 using a direct transcription method called the Gauss pseudospectral method. OPTIM-TOOL 2, on the other hand, is based on the *Legendre pseudospectral method* [40,41]. For brevity, and given the mature level of the Gauss pseudospectral method, the derivation of this method will not be presented in this paper. However, it is important to note that the cost function and constraints that define the nonlinear problem (NLP) are the result of this method, and that the solution of this NLP is an approximate solution to the continuous-time-optimal control problem.

B. Sparse Nonlinear Optimizer

The NLP to be described in Sec. IV was solved with the MATLAB interface of the NLP solver SNOPT using analytical first-order derivatives for the Jacobian and the gradient of the objective function. This NLP solver is a general-purpose system for constrained optimization. It minimizes a linear or nonlinear function subject to bounds on the variables and sparse linear or nonlinear constraints. The NLP solver uses a sequential quadratic programming algorithm [42] to solve the problem. The search direction is obtained through quadratic programming of subproblems that minimize a quadratic model of a Lagrangian function subject to linearized constraints. To ensure that convergence is independent of the starting point, an augmented Lagrangian merit function is reduced along each search direction.

Details regarding the MATLAB implementation of the NLP solver may be found in the user's guide for SNOPT [43]; interested readers may also refer to the work of Gill et al. [39] for details on the algorithm itself. In the context of this paper, it is sufficient to note that this algorithm is suitable for solving nonlinear problems of the following form:

Minimize $\boldsymbol{x}$ for $f(\boldsymbol{x})$ subject to

$$\boldsymbol{l} \le \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{c}(\boldsymbol{x}) \\ \boldsymbol{A}\boldsymbol{x} \end{pmatrix} \le \boldsymbol{u}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is a vector of n variables, $\boldsymbol{l} \in \mathbb{R}$ and $\boldsymbol{u} \in \mathbb{R}$ are constant lower and upper bounds for which the dimensions depend on the number of constraints and variables, $f(\boldsymbol{x}) \in \mathbb{R}$ is a smooth scalar objective function, $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is a sparse matrix, and $\boldsymbol{c}(\boldsymbol{x}) \in \mathbb{R}^m$ is a vector of m smooth nonlinear constraint functions $c_i(\boldsymbol{x}) \in \mathbb{R}$ with sparse derivatives. Upper and lower bounds are specified for all variables and constraints.

Ultimately, the ideal output from the NLP solver is that the optimality conditions are satisfied; in other words, the returned solution $\boldsymbol{x}$ is feasible, the reduced gradient is negligible, the reduced costs are optimal, and the upper-triangular matrix $\boldsymbol{R}$ is nonsingular, which ensures that the sparse matrix $\boldsymbol{A}$ is invertible [39]; note that matrix $\boldsymbol{R}$ is obtained from a dense Cholesky factorization and is referred to as the Cholesky factor of the sparse matrix $\boldsymbol{A}$.

IV. Simulation Results

The simulation results presented in this section uses the FFR in Fig. 1, for which the parameters are defined in Table 1. These parameters represent the properties of the experimental platforms used at Carleton University. The optimization was performed once

Table 1 Parameters of the FFR

Body	Body number	ϕ, deg	a_i, m	b_i, m	m_i, kg	I_i, kg · m^2
Base satellite	0	68.284	— —	0.2304	16.948	2.873×10^{-1}
Link 1	1	— —	0.1933	0.1117	0.3377	3.750×10^{-3}
Link 2	2	— —	0.1993	0.1057	0.3281	3.413×10^{-3}
Link 3	3	— —	0.0621	0.0159	0.0111	5.640×10^{-5}

Table 2 Initial and final kinematic boundaries

	x_0	y_0	q_0	q_1	q_2	q_3	x_w	y_w	x_{ee}	y_{ee}
Initial positions	0 m	0 m	0 rad	$\pi/2$ rad	$\pi/2$ rad	0 rad	— —	— —	— —	— —
Initial velocities	0 m/s	0 m/s	0 rad/s	0 rad/s	0 rad/s	0 rad/s	— —	— —	— —	— —
Final positions	— —	— —	— —	— —	— —	— —	−0.08 m	— —	−0.08 m	0.70 m
Final velocities	0 m/s	0 m/s	0 rad/s	0 rad/s	0 rad/s	0 rad/s	— —	— —	— —	— —

Table 3 Control and path constraints

	F_x, N	F_y, N	τ_0, N/m	τ_1, N	τ_2, N	τ_3, N	q_1, rad	q_2, rad	q_3, rad
Minimum	0	0	0	−0.05	−0.05	−0.05	$-\pi/2$	$-\pi/2$	$-\pi/2$
Maximum	0	0	0	0.05	0.05	0.05	$\pi/2$	$\pi/2$	$\pi/2$

with 30 nodes and a zero initial guess to obtain an initial guess at the solution. This 30-node solution was then used as the initial guess for a 100-node solution in order to refine the optimal trajectories.

A. Optimal Control Problem Setup

The guidance problem solved in this paper is the optimal deployment of a robotic manipulator on a FFR. First, the cost function is defined as the square of the difference between the initial and final attitudes:

$$J(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{\tau}) = (q_0(t_f) - q_0(t_0))^2 \tag{26}$$

This cost function was selected instead of minimizing the overall attitude displacement so that the computation time could be reduced. This reduced computation is the result of not having any running cost function. The constraints for the problem are as follows; first, the problem is subject to the dynamic constraints of the FFR:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{\tau} \tag{27}$$

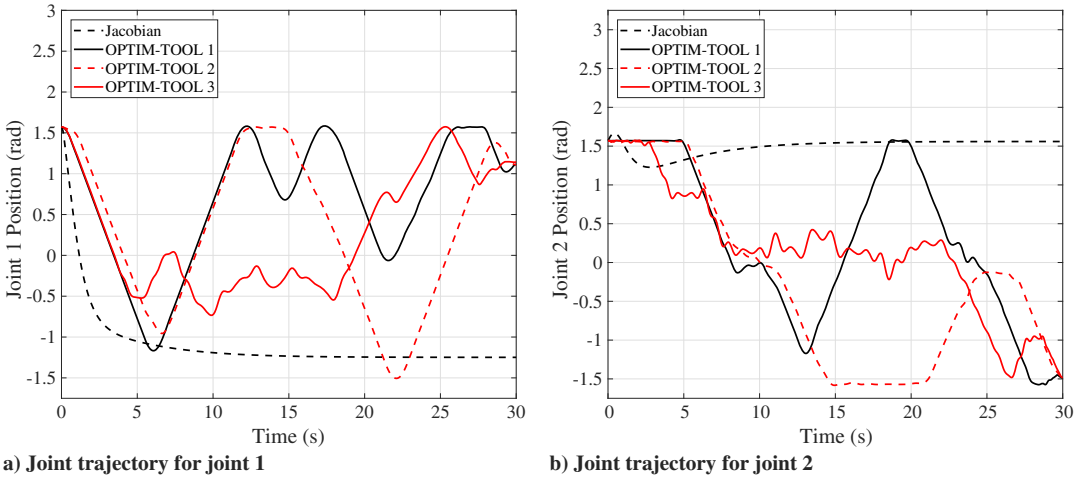
as well as the initial and generalized states and derivatives, as summarized in Table 2. Note that the final end-effector pose is also constrained, as well as the final wrist pose, thereby ensuring that the FFR is in the ideal capture position. The final EE pose is implicitly constrained by Eqs. (6) and (7).

Additionally, the controls and states are subject to upper and lower bounds as summarized in Table 3; it is of particular importance that the path constraints on the states are selected based on the physical limits of the FFR used at Carleton University, and the control constraints are selected to ensure that the motion of the robotic manipulator is smooth; note that the control constraints imposed on the body forces and torque are used to ensure the system is free floating. Any states and controls no listed in Table 3 are to be considered unconstrained.

B. Simulation Results

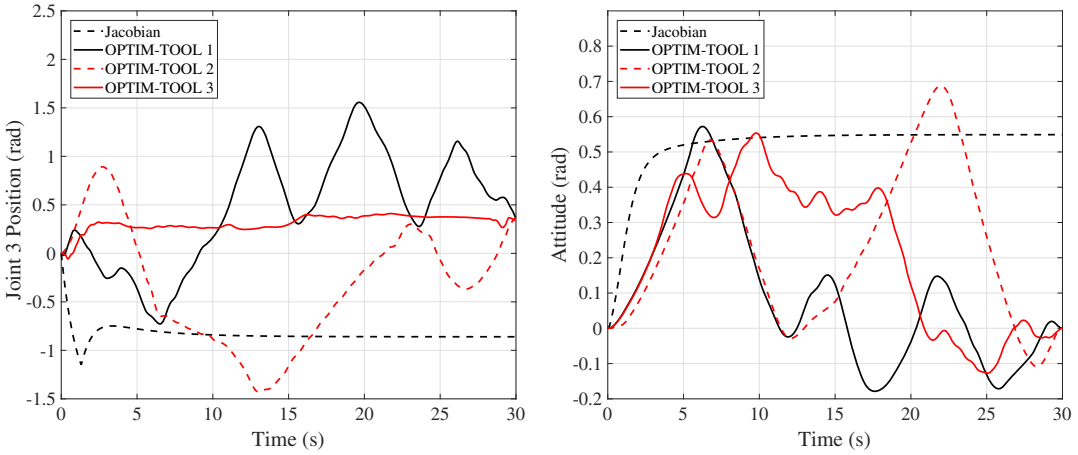
The simulations were performed on a Windows computer with an Intel Core i7-7700HQ CPU at 2.80 GHz. As the experiments were be performed using joint position control, only the optimized joint positions in Figs. 2 and 3 will be shown. For the transpose Jacobian controller, the gains were chosen to be $\boldsymbol{K}_P = 500$, $\boldsymbol{K}_I = 0$, and $\boldsymbol{K}_D = 50$. The joint trajectories in Figs. 2 and 3 result in the EE trajectories found in Figs. 4 and 5.

Based solely on the results presented in Figs. 2–5, it could be concluded that each optimization tool successfully converged to a solution that minimized the attitude disturbance by the end of the maneuver. The fact that each solution is unique also supports the



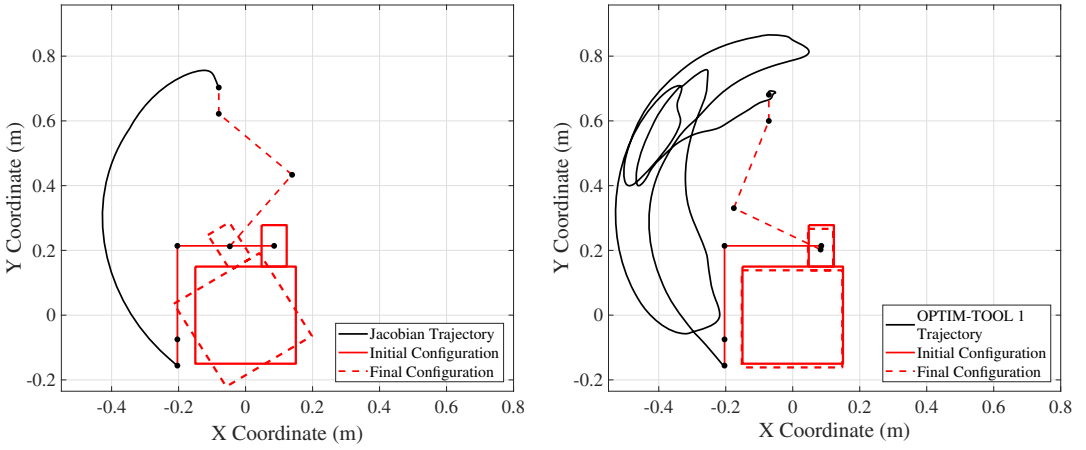
a) Joint trajectory for joint 1 **b) Joint trajectory for joint 2**

Fig. 2 Optimal joint trajectories for joints 1 (left) and 2 (right).



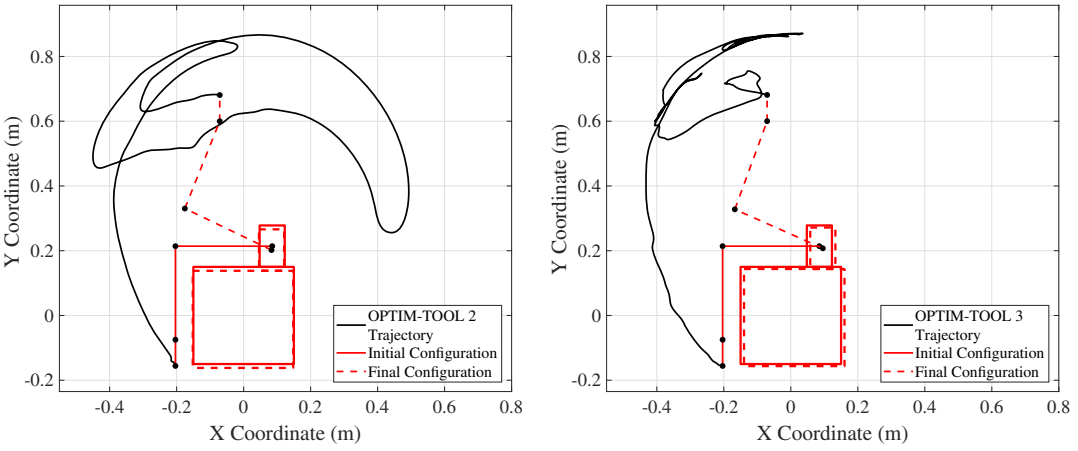
a) Joint trajectory for joint 3 **b) Attitude disturbance as a result of optimal trajectories**

Fig. 3 Optimal joint trajectory for joint 3 (left) and overall attitude disturbance (right).



a) End-effector trajectory obtained using transpose Jacobian controller **b) End-effector trajectory obtained using OPTIM-TOOL 1**

Fig. 4 End-effector trajectories obtained using the transpose Jacobian controller (left) and OPTIM-TOOL 1 (right).



a) End-effector trajectory obtained using OPTIM-TOOL 2 **b) End-effector trajectory obtained using OPTIM-TOOL 3**

Fig. 5 End-effector trajectories obtained using OPTIM-TOOL 2 (left) and OPTIM-TOOL 3 (right).

conclusion that this NLP has several local minima: all of which result in reduced disturbances. Recall that the optimization was first performed with 30 nodes and no initial guess. The 30-node solution was then used as an initial guess to obtain a 100-node solution. The convergence time for both cases is provided in Table 4, but only the final attitude for the 100-node case is presented.

By comparing these results to those obtained by Flores-Abad et al. [16], who solved a similar optimization problem using TOMLAB, it can be observed that the final attitude displacement has been greatly reduced. However, it is important to note that the optimal solution obtained by Flores-Abad et al. [16] was the result of a different cost function. The advantages present in their work included a smaller maximum transient attitude displacement, as well as generally smoother motion and smaller joint torques. This comes as a direct result of their chosen cost function, which was the torque applied by the manipulator on the body of the spacecraft. In contrast, the solution

Table 4 Convergence time and final attitude for each optimization solution

Controller	Convergence time, s 30 nodes with no guess	Convergence time, s 100 nodes with guess	Final attitude, rad 100 nodes with guess
Jacobian	— —	— —	0.5606
OPTIM-TOOL 1	17.425	51.975	≈0
OPTIM-TOOL 2	15.765	46.846	≈0
OPTIM-TOOL 3	10.723	24.334	≈0

presented in this paper provides an optimal trajectory that ensures the spacecraft body returns to its starting attitude. As a consequence, the maximum attitude displacement during the robotic deployment is greater, and the overall trajectories are more complex and require more energy to perform. It is also interesting to note the differences between the optimal solutions and the baseline transpose Jacobian case. The trajectory generated by the Jacobian controller is much simpler, which makes sense, given how it is analogous to a proportional-integral-derivative (PID) controller. By extension, the transpose Jacobian trajectory results in reduced energy consumption when compared to the longer trajectories generated by the optimization tools. However, the consequence is that the attitude deviates significantly. On orbit, this would typically have to be corrected using thrusters or reaction wheels; however, using the technique in this paper, this attitude deviation can be completely avoided while still ensuring that the EE achieves the desired pose.

V. Verification of Optimal Solutions

The results obtained from each tool are first checked using Pontryagin's minimization principle, and then by using Bellman's principle of optimality. The dynamic fidelity of the solutions is then verified by propagating the states, and the conservation of angular momentum is verified.

A. Necessary Conditions

To be considered optimal, any candidate solution must satisfy Pontryagin's necessary conditions for optimality. The theory for these conditions will be summarized here; however, a more thorough explanation can be found in the work of Ross [37]. First, the Hamiltonian $H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}, t) \in \mathbb{R}$ is defined to be a function of the running cost $F(\boldsymbol{x}, \boldsymbol{u}, t) \in \mathbb{R}$, the vector of costates $\boldsymbol{\lambda} \in \mathbb{R}^n$, and the right-hand side of the dynamics $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t)$, as shown in Eq. (28). For a given control function $\boldsymbol{u}$ to be considered a candidate optimal solution, it is necessary that it globally minimizes the Hamiltonian for all time. Put another way, minimizing the Hamiltonian with respect to the control function (while ensuring that $\boldsymbol{\lambda}$ and $\boldsymbol{x}$ are constants) will yield a potential solution for the optimal control problem:

$$H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}, t) = F(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) \tag{28}$$

The path constraints must also be taken into account. In this case, the sole path constraints are the limits on the control and state vectors as defined in the previous section. The Lagrangian of the Hamiltonian can then be defined as follows:

$$\bar{H}(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}, t) = H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\mu}^T \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, t) \tag{29}$$

where the parameters in $\boldsymbol{\mu} \in \mathbb{R}^m$ are the covectors corresponding to the m-path constraints on the states and controls. The lower Hamiltonian $\mathcal{H}(\boldsymbol{\lambda}, \boldsymbol{x}, t) \in \mathbb{R}$ is the minimized Hamiltonian with respect to the controls, and it can be expressed by the following:

$$\mathcal{H}(\boldsymbol{\lambda}, \boldsymbol{x}, t) = \min\, H(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{u}, t) \tag{30}$$

Finally, to account for the constraints on the start and end points of the optimal control problem, one must define the endpoint Lagrangian for the following problem:

$$\bar{E}(t_0, t_f, \boldsymbol{x}_0, \boldsymbol{x}_f, \boldsymbol{v}) = E_c(t_0, t_f, \boldsymbol{x}_0, \boldsymbol{x}_f) + \boldsymbol{v}^T \boldsymbol{e}(t_0, t_f, \boldsymbol{x}_0, \boldsymbol{x}_f) \tag{31}$$

where $E_c(t_0, t_f, \boldsymbol{x}_0, \boldsymbol{x}_f) \in \mathbb{R}$ is the endpoint constraint, $\boldsymbol{v} \in \mathbb{R}^n$ is the n-endpoint covector, and $\boldsymbol{e}(t_0, t_f, \boldsymbol{x}_0, \boldsymbol{x}_f) \in \mathbb{R}^n$ are the n-path constraints. There are a total of five necessary conditions based on the previously defined scalar equations that must be true for any proposed optimal solution. The first necessary condition for optimality is the Hamiltonian minimization condition (HMC), which states the following equality must be true at all points:

$$\frac{\partial \bar{H}}{\partial \boldsymbol{u}} = \boldsymbol{0} \tag{32}$$

Next, the adjoint condition states that the time derivative of the costates can be obtained from the following equation. Note that this condition, for the case presented in this paper, will not provide any meaningful insight into the validity of any solutions because the partial derivatives prove to be large nonlinear equations:

$$-\dot{\boldsymbol{\lambda}} = \frac{\partial \bar{H}}{\partial \boldsymbol{x}} \tag{33}$$

The next condition is known as the transversality condition and, in essence, it ensures that the optimal solution has achieved the desired start and end points. This condition can be expressed as follows:

$$\boldsymbol{\lambda}(t_0) = -\frac{\partial \bar{E}}{\partial \boldsymbol{x}_0} \tag{34}$$

and

$$\boldsymbol{\lambda}(t_f) = \frac{\partial \bar{E}}{\partial \boldsymbol{x}_f} \tag{35}$$

The Hamiltonian value condition then requires that the equality in Eq. (36) be true. For a case in which the final time is unconstrained, this condition indicates that the final value for the lower Hamiltonian should be zero:

$$\mathcal{H}(t_f) = -\frac{\partial \bar{E}}{\partial t_f} = 0 \tag{36}$$

where the states, controls, and costates being fed into the Hamiltonian are the optimal values at the final time. Finally, the Hamiltonian evolution equation requires that the optimal trajectory satisfies the equality in Eq. (37). This equality implies that the lower Hamiltonian should be a constant for all time:

$$\frac{\mathrm{d}\mathcal{H}}{\mathrm{d}t} = 0 \tag{37}$$

Checking each condition does not always provide useful information, however, because certain conditions amount to comparing one unknown to another. Checking a few conditions when possible can be a strong indication that a local minimum has been found. Yet, one must be aware that, if all checked conditions hold true, it does not guarantee that a minimum has been found.

In this paper, validation will be performed by first checking that the Hamiltonian value condition in Eq. (36) is zero (because the endpoint Lagrangian is not directly time dependent). In addition, the transversality condition will be used to check the start and end points of each solution. Finally, knowing that the time derivative of the lower

Hamiltonian must be zero to satisfy the Hamiltonian evolution equation, the lower Hamiltonian can be calculated at each time step and should be a constant for all time. Note that all costates and covectors were obtained through the optimization tools.

1. Hamiltonian Value Condition

Substituting the optimal solution from each tool into Eq. (28) yields the following results:

OPTIM-TOOL 1:

$$\mathcal{H}(t_f) = -7.1735 \times 10^{-9}$$

OPTIM-TOOL 2:

$$\mathcal{H}(t_f) = -3.3144 \times 10^{-9}$$

OPTIM-TOOL 3:

$$\mathcal{H}(t_f) = -4.5379 \times 10^{-8}$$

As the Hamiltonian value at the final node is very close to zero for all tools, it can be stated that the Hamiltonian value condition holds true.

2. Transversality Condition

Next, the transversality conditions in Eqs. (34) and (35) are checked to ensure that the solver converged to solutions that achieved the constraints imposed on the initial and final states, and the results are summarized in Tables 5–7. In these tables, each row represents a component of the transversality check, and each column represents an element of the vector.

The results from the transversality condition are consistent with the observations made using the Hamiltonian value condition, which is to say that all solutions have met the transversality condition. It is interesting to note, however, that certain elements for OPTIM-TOOL 3 are mismatched; that being said, the magnitude of the values renders this mismatch relatively insignificant.

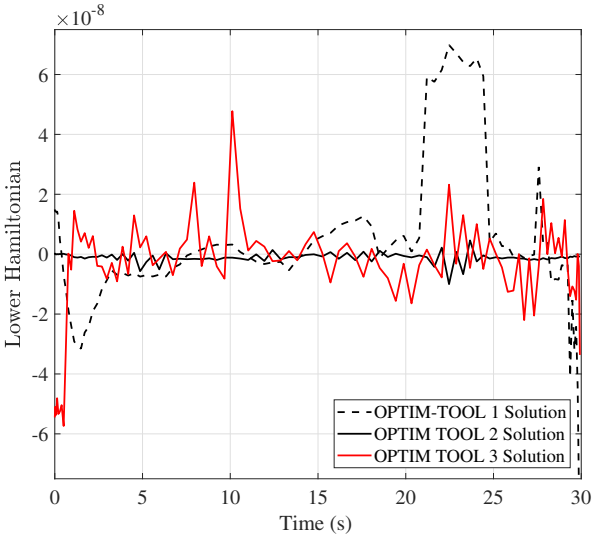


Fig. 6 Hamiltonian evolution equation at each time step.

3. Hamiltonian Evolution Equation

Next, the lower Hamiltonian can be evaluated at each time step to ensure that it is constant, as required by the Hamiltonian evolution equation condition (see Fig. 6).

As was the case for the other verifications performed thus far, the solution obtained by OPTIM-TOOL 2 has the smallest variance in the lower Hamiltonian, which provides further confidence that the solution obtained is an extremal. The solutions from OPTIM-TOOL 1 and OPTIM-TOOL 3 are both very close to constant when consideration is made for the scale of the figure, and they are therefore likely to be extremal solutions.

B. Bellman's Principle of Optimality

Next, Bellman's principle of optimality is checked. In brief, this principle states that any optimal control law will, regardless of the initial state and initial decision, converge to the same optimal policy

Table 5 Transversality condition for OPTIM-TOOL 1

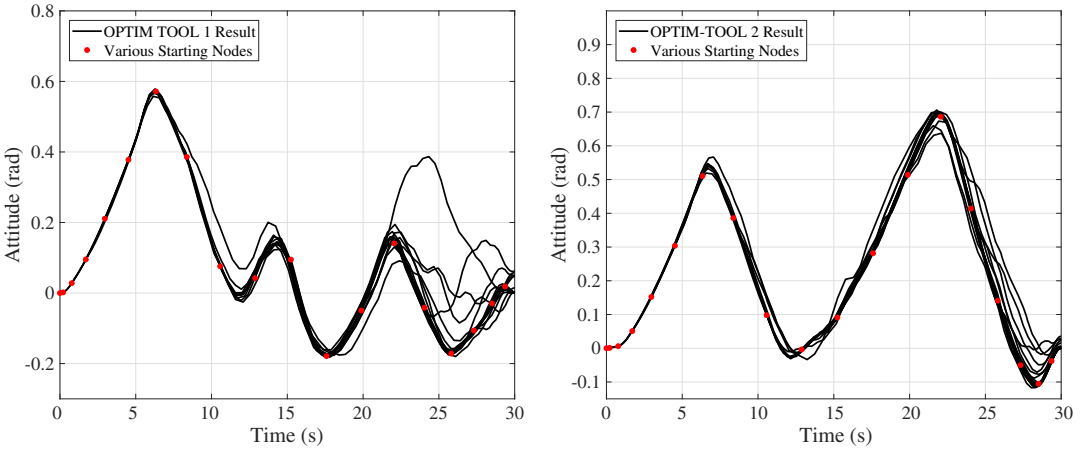
	References											
	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
$\lambda(t_0)(\times 10^{-8})$	0.00	0.00	0.00	−23.8	2.83	−0.01	0.00	0.00	0.00	−1.96	−0.52	−0.01
$\lambda(t_f)(\times 10^{-8})$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.9	15.4	0.26
$-(\partial\bar{E}/\partial \boldsymbol{x}_0)(\times 10^{-8})$	0.00	0.00	0.00	−23.8	2.83	−0.01	0.00	0.00	0.00	−1.96	−0.52	−0.01
$(\partial\bar{E}/\partial \boldsymbol{x}_f)(\times 10^{-8})$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.9	15.4	0.26

Table 6 Transversality condition for OPTIM-TOOL 2

	References											
	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
$\lambda(t_0)(\times 10^{-9})$	0.00	0.00	0.04	17.5	−2.79	−0.24	0.00	0.00	0.00	0.16	0.01	0.00
$\lambda(t_f)(\times 10^{-10})$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	−10.2	−0.72
$-(\partial\bar{E}/\partial \boldsymbol{x}_0)(\times 10^{-9})$	0.00	0.00	0.04	17.5	−2.79	−0.24	0.00	0.00	0.00	0.16	0.01	0.00
$(\partial\bar{E}/\partial \boldsymbol{x}_f)(\times 10^{-10})$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	−10.2	−0.72

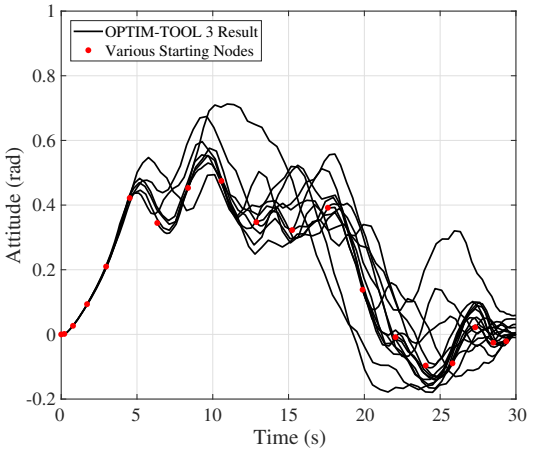
Table 7 Transversality condition for OPTIM-TOOL 3

	References											
	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
$\lambda(t_0)(\times 10^{-9})$	0.00	0.00	0.00	0.00	0.00	0.53	0.00	0.00	0.00	65.7	19.1	0.32
$\lambda(t_f)(\times 10^{-10})$	−210	0.00	132	87.0	60.3	0.00	0.00	0.00	0.00	−47.8	−19.3	−0.40
$-(\partial\bar{E}/\partial \boldsymbol{x}_0)(\times 10^{-9})$	0.00	0.00	0.00	0.00	0.00	0.53	0.00	0.00	0.00	65.7	19.1	0.32
$(\partial\bar{E}/\partial \boldsymbol{x}_f)(\times 10^{-10})$	−210	0	103	76.7	16.4	0.00	0.00	0.00	0.00	−47.8	−19.3	−0.40



a) Applying Bellman's principle to the solution from OPTIM-TOOL 1

b) Applying Bellman's principle to the solution from OPTIM-TOOL 2



c) Applying Bellman's principle to the solution from OPTIM-TOOL 3

Fig. 7 Application of Bellman's principle of optimality toward the solutions from each tool.

[44]. In other words, to check this principle, each solution was run at different starting nodes, and the results are shown in the following. For example, for starting node 20, the optimal control problem was set up such that the initial conditions corresponded to the states at node 20, whereas the final conditions remained the same. If the principle holds true, then the states from nodes 20 to 100 should be the same as the original solution. The results from this study are presented in Fig. 7.

Ultimately, the results from this analysis are consistent with the previous conclusions made using Pontryagin's necessary conditions. In other words, OPTIM-TOOLs 1 and 2 met Bellman's criteria best, followed by OPTIM-TOOL 3. Of particular importance is the spread of results for the validation of the solution from OPTIM-TOOL 3, which is particularly large, further supporting the conclusion that it is the weakest solution.

C. Propagation of Optimal Solutions

In addition to validating the optimality of the numerical results obtained from the tools, it is equally important to validate the dynamic fidelity of the numerical solution using a propagator. To propagate the optimal solution, MATLAB-Simulink was used along with the Runge–Kutta 4(5)th-order ordinary differential equations solver (ODE45) with relative and absolute tolerances set to 10^{-9}. Using the first data points from each optimal solution, each solution was propagated forward to obtain a dataset at a higher sampling rate than the original 100 nodes. Similar to what was done for Bellman's principle of optimality, only the attitude disturbance over time will be shown, and these results are presented in Fig. 8. The best solution will be the one that has the smallest deviation from the propagated attitude.

From these results, it can be noted the states estimated by OPTIM-TOOL 3 do not match the propagated results as well as the other two solutions. Despite the lower fidelity in the solution from OPTIM-TOOL 3, the trajectories from each tool will be used as the desired joint angles for the actuators during the experimental validation campaign, as outlined in the following section. By using results that are known to be truly optimal in addition to one solution that is known to be inconsistent in terms of its dynamic fidelity, this paper will demonstrate the importance of validating the solutions obtained from these so-called "black-box" tools.

D. Conservation of Angular Momentum

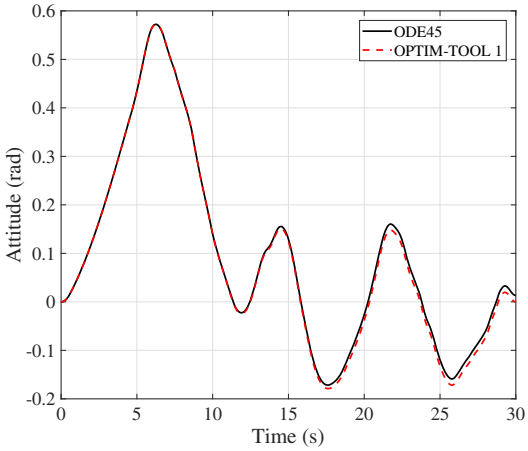
Finally, one of the primary assumptions made in this paper is that the total angular momentum of the system is conserved. The total angular momentum of the system for a planar case can be calculated using the following [24]:

$$\boldsymbol{P}_{\text{sys}} = \tilde{\boldsymbol{H}}_{bm}\dot{\boldsymbol{q}} \tag{38}$$

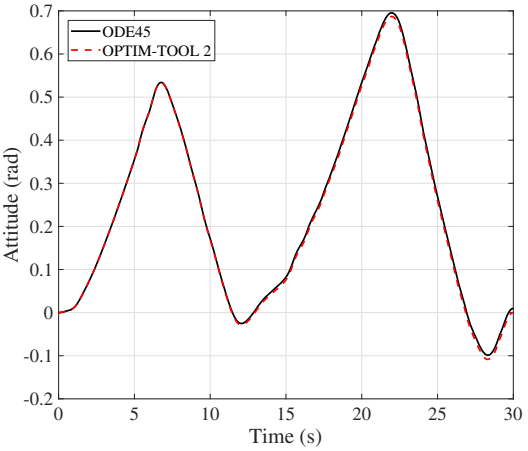
For the simulation results, the total angular momentum of the system for each solution can be plotted as a function of time, as shown in Fig. 9; if momentum is conserved, then the momentum at $t = 0$ should be the same as at $t = t_f$. From the results in Fig. 9, it can be concluded that the angular momentum has been conserved in simulation.

VI. Experimental Validation

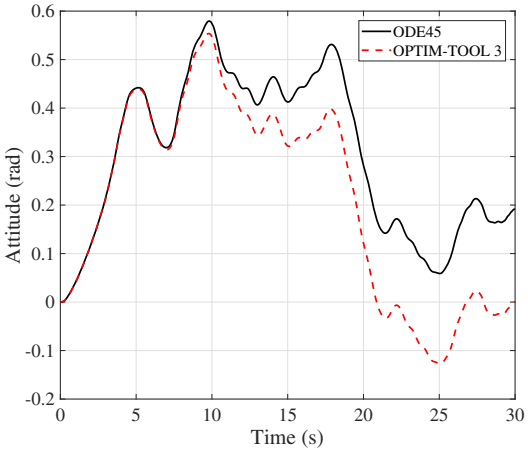
The facility at Carleton University has two main components: a large gravity offset table, and two free-floating spacecraft simulator



a) Propagation of the solution from OPTIM-TOOL 1



b) Propagation of the solution from OPTIM-TOOL 2



c) Propagation of the solution from OPTIM-TOOL 3

Fig. 8 Propagation of the solutions from each tool using a numerical propagator.

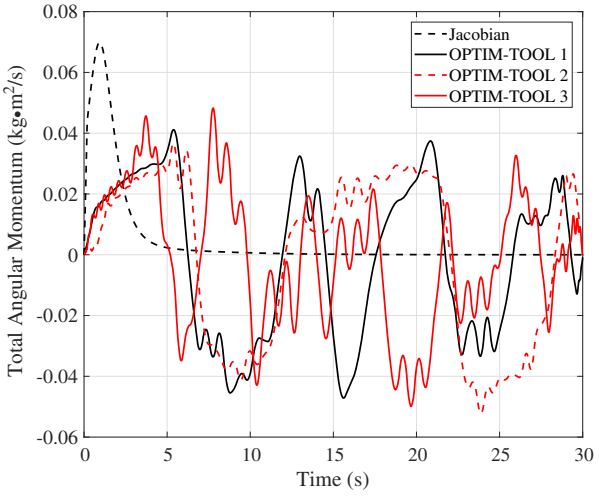


Fig. 9 Angular momentum for the system.

Fig. 10 SPOT FFR and robotic manipulator.

platforms. The red platform used in experiments is shown in Fig. 10, in addition to the gravity offset table and the manipulator. The platforms use compressed air and air bearings to create a near-frictionless environment. A laboratory computer is used to upload the guidance, navigation, and control (GN&C) software; initiate experiments; and download the experimental data at the end of each test [45].

The platforms use a Raspberry Pi 3 (RP3), running a Raspbian Jessie Linux distribution, to execute control commands as well as log data and provide an interface for the various instruments on board. The GN&C software is first designed in MATLAB-Simulink, in which simulations can be performed before running an experiment. Following the simulations, the Simulink diagram is exported to C++ code using the Embedded Coder toolbox; the generated code is then compiled into a Linux executable on board each active platform. Any instrumentation on the platform is integrated into the GN&C software using custom Simulink driver blocks.

The platforms are capable of performing a multitude of different experiments, such as formation flying maneuvers, stabilization of noncooperative targets, and robotic manipulation tasks. Several experiments have already been performed using the experimental testbed, including an experimental validation for tethered capture of a spinning space debris [45,46] as well as iterative learning control of spacecraft proximity operations based on confidence level [47].

To determine the position of the platforms during an experiment, an array of eight PhaseSpace motion capture cameras track the inertial position of light-emitting diodes (LEDs), each with their own unique pulse frequency. The reported resolution of this camera system is 0.01 mm; and position measurements are made in the x, y, and z directions. Both platforms have four LEDs, with one located at each corner of the platform's top panel. Using these, the PhaseSpace software can derive the quaternions of the rigid body to provide a measurement of the attitude, as well as track the positions of the spacecraft in inertial space. There are also three LEDs on the robotic manipulator, which are used to track the positions of the elbow joint, the wrist joint, and the EE in inertial space. The robotic manipulator attached to the red platform consists of three links and three actuators (Dynamixel MX-64), with a maximum reach of approximately 65.88 cm. Each actuator is capable of a maximum 6.0 N/m of torque at 12 V.

A. Experimental Procedure

To experimentally validate the results obtained in Sec. IV, the following experimental test plan was conceived. It is important to first note that experimental validation of the optimal guidance technique is performed in an open-loop fashion; in other words, the optimal joint positions determined in simulation were supplied directly to the actuators, which house their own (tunable) PID joint position controllers. The control logic is shown in Fig. 11.

The experimental validation will be considered a success if the perturbed attitude of the platform is similar to the disturbance observed in simulation. In the case of the solution from TOMLAB, it has already been determined that the dynamic fidelity is poor, and it is therefore expected that the experiment results will be much closer to the results obtained when propagating the dynamics model (as opposed to the attitude estimated by TOMLAB itself). To ensure the repeatability of the results, each experiment was performed 10 times. The mean of the trials is shown in the figures present in the following subsection, along with error bars representing the 99.7% (i.e., 3σ confidence interval for the trials) and the expected attitude obtained from each tool; note that the error bars are calculated using the standard error. The gains for the MX-64 PID controller were chosen to be $K_P = 400$, $K_I = 0$, and $K_D = 200$.

B. Results

For the experimental results shown in the following, a video of each experiment is available at https://youtu.be/8RI49Z7BpIQ; these experiment videos are each overlaid with videos showing their respective simulation results. The experimental results are presented in Figs. 12 and 13.

The results obtained compare favorably with the simulated results, with the exception of OPTIM-TOOL 3. However, in the case of OPTIM-TOOL 3, the attitude deviation observed in the experiment is very similar to the results obtained when propagating the solution using ODE45. This further supports the fact that the solutions obtained from any optimization tool must be validated. Without the analysis presented in Sec. V, a user would have no indication that their optimal solution was, in fact, suboptimal or even infeasible.

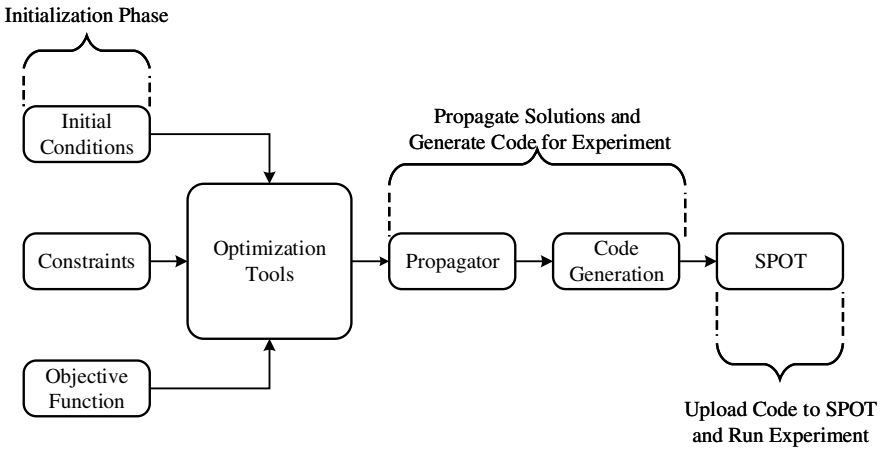
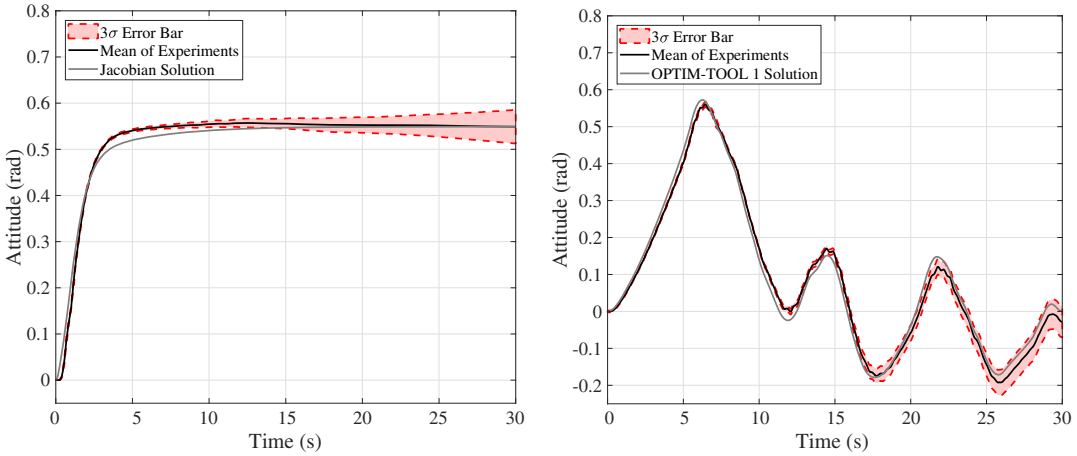


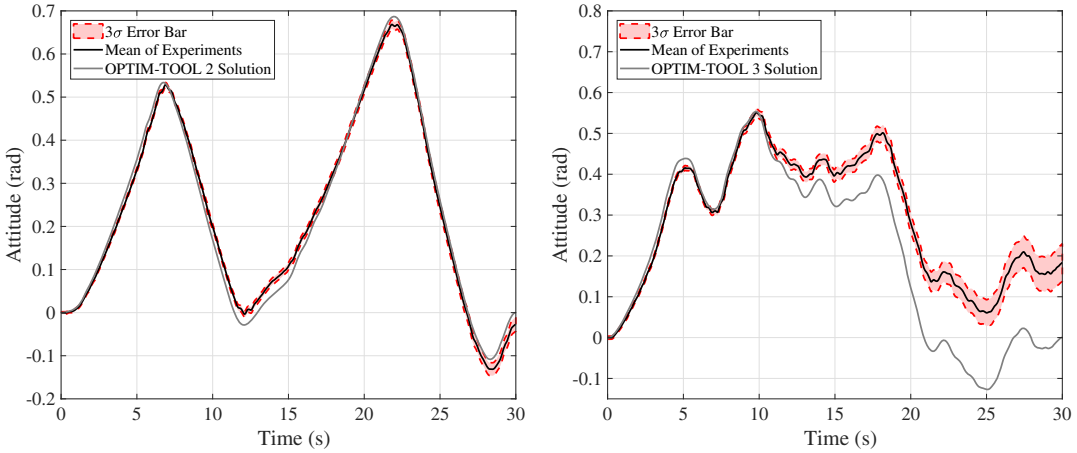
Fig. 11 Control logic for experiment.



a) Attitude disturbance observed in experiment for the transpose Jacobian controller

b) Attitude disturbance observed in experiment for the solution from OPTIM-TOOL 1

Fig. 12 Attitude disturbance observed for the transpose Jacobian (left) and OPTIM-TOOL 1 (right).



a) Attitude disturbance observed in experiment for the solution from OPTIM-TOOL 2

b) Attitude disturbance observed in experiment for the solution from OPTIM-TOOL 3

Fig. 13 Attitude disturbance observed for OPTIM-TOOL 2 (left) and OPTIM-TOOL 3 (right).

It is important to address the sources of error and assumptions made in this paper. In no particular order, the following items all contributed to errors observed in the experiments:

1) The mass properties for the platform do not account for the change in mass associated with air used for the air bearings and thrusters, which amounts to approximately 0.5 kg when going from a full tank to an empty tank of compressed air. The assumption that the inertia of the platform remains constant is likely one of the main contributing factors to the drifting error that can be observed in the experimental results. Because the platform inertia is actually changing with each experiment, it is expected that the attitude will drift away from the simulation.

2) The robotic manipulator on the FFR is currently self-supported, and although this does provide some benefits in terms of ease of use, it also reduces the validity of the rigid-body assumption made in the derivation of the dynamics model. This vertical flex in the linkages means that, for the same joint angles, the manipulator will not reach the same end-effector coordinate expected from simulation.

3) The facilities at the SRCL are not kept in a clean room, and dust accumulation is a serious hindrance during experiments. If careful preparations are not made before each trial, large drifts in attitude have been observed; these drifts have been attributed to the air bearings getting stuck momentarily on dust particles. Before each trial, a lengthy cleaning process was followed, and thus this problem should not be a major contributor to the experimental error.

4) To validate the solutions obtained from each tool, the PID gains on the position controller were tuned in such a way that the actuators would follow the desired joint angles very closely. However, this workaround does induce errors because it is not possible for the PID controller to follow the joint trajectories perfectly.

VII. Conclusions

In this paper, nonlinear pseudospectral-based optimal guidance techniques for free-floating robots were validated in simulations and experiments. This was accomplished by first deriving the kinematic and dynamic equations for a free-floating robot with a three-degree-of-freedom robotic manipulator that described the motion of the end effector as a function of the generalized joint coordinates. Next, optimal trajectory planning techniques were presented and a cost function was defined. Then, simulations were used to demonstrate the performance of the techniques, and the results showed a reduction in attitude disturbance. The extremal solutions provided by each tool were verified using Pontryagin's minimization principle as well as Bellman's principle of optimality, and it was determined that OPTIM-TOOLs 1 and 2 provided the best solution in terms of optimality and dynamic fidelity. Finally, experimental validation was performed, and the results demonstrated that the optimal trajectory ensured that the free-floating robot was able to deploy the manipulator while still returning to its initial attitude by the end of the maneuver.

Appendix: Elements of the Inertia and Coriolis Matrices

The elements $M_{ij} \forall i = 1 \ldots 6, j = 1 \ldots 6$ are defined as follows:

$$
\begin{aligned}
M_{11} &= m_0 + m_1 + m_2 + m_3 \\
M_{12} &= 0 \\
M_{13} &= (-a_1 m_1 - m_2(a_1 + b_1) - m_3(a_1 + b_1)) \cos(q_0 + q_1) \\
&\quad + (-a_2 m_2 - m_3(a_2 + b_2)) \cos(q_0 + q_1 + q_2) \\
&\quad + (-a_3 m_3) \cos(q_0 + q_1 + q_2 + q_3) \\
&\quad + (-b_0 m_1 - b_0 m_2 - b_0 m_3) \sin(\phi + q_0) \\
M_{14} &= (-a_1 m_1 - m_2(a_1 + b_1) - m_3(a_1 + b_1)) \cos(q_0 + q_1) \\
&\quad + (-a_2 m_2 - m_3(a_2 + b_2)) \cos(q_0 + q_1 + q_2) \\
&\quad + (-a_3 m_3) \cos(q_0 + q_1 + q_2 + q_3) \\
M_{15} &= (-a_2 m_2 - a_2 m_3 - b_2 m_3) \cos(q_0 + q_1 + q_2) \\
&\quad + (-a_3 m_3) \cos(q_0 + q_1 + q_2 + q_3) \\
M_{16} &= (-a_3 m_3) \cos(q_0 + q_1 + q_2 + q_3) \\
M_{21} &= 0 \\
M_{22} &= m_0 + m_1 + m_2 + m_3 \\
M_{23} &= (b_0 m_1 + b_0 m_2 + b_0 m_3) \cos(\phi + q_0) \\
&\quad + (-a_1 m_1 - m_2(a_1 + b_1) - m_3(a_1 + b_1)) \sin(q_0 + q_1) \\
&\quad + (-a_2 m_2 - m_3(a_2 + b_2)) \sin(q_0 + q_1 + q_2) \\
&\quad + (-a_3 m_3) \sin(q_0 + q_1 + q_2 + q_3) \\
M_{24} &= (-a_1 m_1 - m_2(a_1 + b_1) - m_3(a_1 + b_1)) \sin(q_0 + q_1) \\
&\quad + (-a_2 m_2 - m_3(a_2 + b_2)) \sin(q_0 + q_1 + q_2) \\
&\quad + (-a_3 m_3) \sin(q_0 + q_1 + q_2 + q_3) \\
M_{25} &= (-a_2 m_2 - a_2 m_3 - b_2 m_3) \sin(q_0 + q_1 + q_2) \\
&\quad + (-a_3 m_3) \sin(q_0 + q_1 + q_2 + q_3) \\
M_{26} &= (-a_3 m_3) \sin(q_0 + q_1 + q_2 + q_3) \\
M_{31} &= M_{13} \\
M_{32} &= M_{23}
\end{aligned}
$$

$$M_{33} = (2a_1a_2m_2 + 2a_1a_2m_3 + 2a_2b_1m_2 + 2a_1b_2m_3 + 2a_2b_1m_3 + 2b_1b_2m_3)\cos(q_2) + (2a_2a_3m_3 + 2a_3b_2m_3)\cos(q_3) + (2a_1a_3m_3 + 2a_3b_1m_3)\cos(q_2 + q_3) + (2a_1b_0m_1 + 2a_1b_0m_2 + 2a_1b_0m_3 + 2b_0b_1m_2 + 2b_0b_1m_3)\sin(\phi - q_1) + (-2a_2b_0m_2 - 2a_2b_0m_3 - 2b_0b_2m_3)\sin(q_1 - \phi + q_2) + (-2a_3b_0m_3)\sin(q_1 - \phi + q_2 + q_3) + I_0 + I_1 + I_2 + I_3 + a_1{}^2m_1 + a_1{}^2m_2 + a_1{}^2m_3 + a_2{}^2m_2 + a_2{}^2m_3 + a_3{}^2m_3 + b_0{}^2m_1 + b_0{}^2m_2 + b_0{}^2m_3 + b_1{}^2m_2 + b_1{}^2m_3 + b_2{}^2m_3 + 2a_1b_1m_2 + 2a_1b_1m_3 + 2a_2b_2m_3$$

$$M_{34} = (2a_1a_2m_2 + 2a_1a_2m_3 + 2a_2b_1m_2 + 2a_1b_2m_3 + 2a_2b_1m_3 + 2b_1b_2m_3)\cos(q_2) + (2a_2a_3m_3 + 2a_3b_2m_3)\cos(q_3) + (2a_1a_3m_3 + 2a_3b_1m_3)\cos(q_2 + q_3) + (a_1b_0m_1 + a_1b_0m_2 + a_1b_0m_3 + b_0b_1m_2 + b_0b_1m_3)\sin(\phi - q_1) + (-a_2b_0m_2 - a_2b_0m_3 - b_0b_2m_3)\sin(q_1 - \phi + q_2) + (-a_3b_0m_3)\sin(q_1 - \phi + q_2 + q_3) + I_1 + I_2 + I_3 + a_1{}^2m_1 + a_1{}^2m_2 + a_1{}^2m_3 + a_2{}^2m_2 + a_2{}^2m_3 + a_3{}^2m_3 + b_1{}^2m_2 + b_1{}^2m_3 + b_2{}^2m_3 + 2a_1b_1m_2 + 2a_1b_1m_3 + 2a_2b_2m_3$$

$$M_{35} = (a_1a_2m_2 + a_1a_2m_3 + a_2b_1m_2 + a_1b_2m_3 + a_2b_1m_3 + b_1b_2m_3)\cos(q_2) + (2a_2a_3m_3 + 2a_3b_2m_3)\cos(q_3) + (a_1a_3m_3 + a_3b_1m_3)\cos(q_2 + q_3) + (-a_2b_0m_2 - a_2b_0m_3 - b_0b_2m_3)\sin(q_1 - \phi + q_2) + (-a_3b_0m_3) \times \sin(q_1 - \phi + q_2 + q_3) + I_2 + I_3 + a_2{}^2m_2 + a_2{}^2m_3 + a_3{}^2m_3 + b_2{}^2m_3 + 2a_2b_2m_3$$

$$M_{36} = (a_2a_3m_3 + a_3b_2m_3)\cos(q_3) + (a_1a_3m_3 + a_3b_1m_3) \times \cos(q_2 + q_3) + (-a_3b_0m_3)\sin(q_1 - \phi + q_2 + q_3) + m_3a_3{}^2 + I_3$$

$$M_{41} = M_{14}$$

$$M_{42} = M_{24}$$

$$M_{43} = M_{34}$$

$$M_{44} = (2a_1a_2m_2 + 2a_1a_2m_3 + 2a_2b_1m_2 + 2a_1b_2m_3 + 2a_2b_1m_3 + 2b_1b_2m_3)\cos(q_2) + (2a_2a_3m_3 + 2a_3b_2m_3)\cos(q_3) + (2a_1a_3m_3 + 2a_3b_1m_3)\cos(q_2 + q_3) + I_1 + I_2 + I_3 + a_1{}^2m_1 + a_1{}^2m_2 + a_1{}^2m_3 + a_2{}^2m_2 + a_2{}^2m_3 + a_3{}^2m_3 + b_1{}^2m_2 + b_1{}^2m_3 + b_2{}^2m_3 + 2a_1b_1m_2 + 2a_1b_1m_3 + 2a_2b_2m_3$$

$$M_{45} = (a_1a_2m_2 + a_1a_2m_3 + a_2b_1m_2 + a_1b_2m_3 + a_2b_1m_3 + b_1b_2m_3)\cos(q_2) + (2a_2a_3m_3 + 2a_3b_2m_3)\cos(q_3) + (a_1a_3m_3 + a_3b_1m_3)\cos(q_2 + q_3) + I_2 + I_3 + a_2{}^2m_2 + a_2{}^2m_3 + a_3{}^2m_3 + b_2{}^2m_3 + 2a_2b_2m_3$$

$$M_{46} = (a_2a_3m_3 + a_3b_2m_3)\cos(q_3) + (a_1a_3m_3 + a_3b_1m_3) \times \cos(q_2 + q_3) + m_3a_3{}^2 + I_3$$

$$M_{51} = M_{15}$$

$$M_{52} = M_{25}$$

$$M_{53} = M_{35}$$

$$M_{54} = M_{45}$$

$$M_{55} = (2a_2a_3m_3 + 2a_3b_2m_3)\cos(q_3) + I_2 + I_3 + a_2{}^2m_2 + a_2{}^2m_3 + a_3{}^2m_3 + b_2{}^2m_3 + 2a_2b_2m_3$$

$$M_{56} = (a_2a_3m_3 + a_3b_2m_3)\cos(q_3) + m_3a_3{}^2 + I_3$$

$$M_{61} = M_{16}$$

$$M_{62} = M_{26}$$

$$M_{63} = M_{36}$$

$$M_{64} = M_{46}$$

$$M_{65} = M_{56}$$

$$M_{66} = m_3a_3{}^2 + I_3$$

From these components, as described in the body of the paper, the components $C_{ij} \forall i = 1 \ldots 6, j = 1 \ldots 6$ are derived and presented as follows:

$$C_{11} = 0$$

$$C_{12} = 0$$

$$C_{13} = (-\dot{q}_0(b_0m_1 + b_0m_2 + b_0m_3))\cos(\phi + q_0) + (-\dot{q}_0(a_1m_1 + m_2(a_1 + b_1) + m_3(a_1 + b_1)) - \dot{q}_1(a_1m_1 + m_2(a_1 + b_1) + m_3(a_1 + b_1)))\cos\left(q_0 + q_1 + \frac{\pi}{2}\right) + (-\dot{q}_0(a_2m_2 + m_3(a_2 + b_2)) - \dot{q}_1(a_2m_2 + m_3(a_2 + b_2)) - \dot{q}_2(a_2m_2 + m_3(a_2 + b_2)))\cos\left(q_0 + q_1 + q_2 + \frac{\pi}{2}\right) + (-a_3m_3\dot{q}_0 - a_3m_3\dot{q}_1 - a_3m_3\dot{q}_2 - a_3m_3\dot{q}_3) \times \cos\left(q_0 + q_1 + q_2 + q_3 + \frac{\pi}{2}\right)$$

$$C_{14} = (-\dot{q}_0(a_1m_1 + m_2(a_1 + b_1) + m_3(a_1 + b_1)) - \dot{q}_1(a_1m_1 + m_2(a_1 + b_1) + m_3(a_1 + b_1)))\cos\left(q_0 + q_1 + \frac{\pi}{2}\right) + (-\dot{q}_0(a_2m_2 + m_3(a_2 + b_2)) - \dot{q}_1(a_2m_2 + m_3(a_2 + b_2)) - \dot{q}_2(a_2m_2 + m_3(a_2 + b_2)))\cos\left(q_0 + q_1 + q_2 + \frac{\pi}{2}\right) + (-a_3m_3\dot{q}_0 - a_3m_3\dot{q}_1 - a_3m_3\dot{q}_2 - a_3m_3\dot{q}_3) \times \cos\left(q_0 + q_1 + q_2 + q_3 + \frac{\pi}{2}\right)$$

$$C_{15} = (-\dot{q}_0(a_2m_2 + m_3(a_2 + b_2)) - \dot{q}_1(a_2m_2 + m_3(a_2 + b_2)) - \dot{q}_2(a_2m_2 + m_3(a_2 + b_2)))\cos\left(q_0 + q_1 + q_2 + \frac{\pi}{2}\right) + (-a_3m_3\dot{q}_0 - a_3m_3\dot{q}_1 - a_3m_3\dot{q}_2 - a_3m_3\dot{q}_3) \times \cos\left(q_0 + q_1 + q_2 + q_3 + \frac{\pi}{2}\right)$$

$$C_{16} = (a_3m_3(\dot{q}_0 + \dot{q}_1 + \dot{q}_2 + \dot{q}_3))\sin(q_0 + q_1 + q_2 + q_3)$$

$$C_{21} = 0$$

$$C_{22} = 0$$

$$\begin{aligned}C_{23} &= (-\dot{q}_0(b_0m_1 + b_0m_2 + b_0m_3))\sin(\phi + q_0)\\ &+ (-\dot{q}_0(a_1m_1 + m_2(a_1 + b_1) + m_3(a_1 + b_1))\\ &- \dot{q}_1(a_1m_1 + m_2(a_1 + b_1) + m_3(a_1 + b_1)))\sin\left(q_0 + q_1 + \frac{\pi}{2}\right)\\ &+ (-\dot{q}_0(a_2m_2 + m_3(a_2 + b_2)) - \dot{q}_1(a_2m_2 + m_3(a_2 + b_2))\\ &- \dot{q}_2(a_2m_2 + m_3(a_2 + b_2)))\sin\left(q_0 + q_1 + q_2 + \frac{\pi}{2}\right)\\ &+ (-a_3m_3\dot{q}_0 - a_3m_3\dot{q}_1 - a_3m_3\dot{q}_2 - a_3m_3\dot{q}_3)\\ &\times \sin\left(q_0 + q_1 + q_2 + q_3 + \frac{\pi}{2}\right)\end{aligned}$$

$$\begin{aligned}C_{24} &= (-\dot{q}_0(a_1m_1 + m_2(a_1 + b_1) + m_3(a_1 + b_1))\\ &- \dot{q}_1(a_1m_1 + m_2(a_1 + b_1) + m_3(a_1 + b_1)))\sin\left(q_0 + q_1 + \frac{\pi}{2}\right)\\ &+ (-\dot{q}_0(a_2m_2 + m_3(a_2 + b_2)) - \dot{q}_1(a_2m_2 + m_3(a_2 + b_2))\\ &- \dot{q}_2(a_2m_2 + m_3(a_2 + b_2)))\sin\left(q_0 + q_1 + q_2 + \frac{\pi}{2}\right)\\ &+ (-a_3m_3\dot{q}_0 - a_3m_3\dot{q}_1 - a_3m_3\dot{q}_2 - a_3m_3\dot{q}_3)\\ &\times \sin\left(q_0 + q_1 + q_2 + q_3 + \frac{\pi}{2}\right)\end{aligned}$$

$$\begin{aligned}C_{25} &= (-\dot{q}_0(a_2m_2 + m_3(a_2 + b_2)) - \dot{q}_1(a_2m_2 + m_3(a_2 + b_2))\\ &- \dot{q}_2(a_2m_2 + m_3(a_2 + b_2)))\sin\left(q_0 + q_1 + q_2 + \frac{\pi}{2}\right)\\ &+ (-a_3m_3\dot{q}_0 - a_3m_3\dot{q}_1 - a_3m_3\dot{q}_2 - a_3m_3\dot{q}_3)\\ &\times \sin\left(q_0 + q_1 + q_2 + q_3 + \frac{\pi}{2}\right)\end{aligned}$$

$$C_{26} = (-a_3m_3(\dot{q}_0 + \dot{q}_1 + \dot{q}_2 + \dot{q}_3))\cos(q_0 + q_1 + q_2 + q_3)$$

$$C_{31} = 0$$

$$C_{32} = 0$$

$$\begin{aligned}C_{33} &= (-a_1b_0m_1\dot{q}_1 - a_1b_0m_2\dot{q}_1 - a_1b_0m_3\dot{q}_1 - b_0b_1m_2\dot{q}_1\\ &- b_0b_1m_3\dot{q}_1)\cos(\phi - q_1) + (-a_2b_0m_2\dot{q}_1 - a_2b_0m_2\dot{q}_2\\ &- a_2b_0m_3\dot{q}_1 - a_2b_0m_3\dot{q}_2 - b_0b_2m_3\dot{q}_1 - b_0b_2m_3\dot{q}_2)\\ &\times \cos(q_1 - \phi + q_2) + (-a_3b_0m_3\dot{q}_1 - a_3b_0m_3\dot{q}_2 - a_3b_0m_3\dot{q}_3)\\ &\times \cos(q_1 - \phi + q_2 + q_3) + (-a_2b_1m_2\dot{q}_2 - a_1b_2m_3\dot{q}_2\\ &- a_2b_1m_3\dot{q}_2 - b_1b_2m_3\dot{q}_2 - a_1a_2m_2\dot{q}_2 - a_1a_2m_3\dot{q}_2)\sin(q_2)\\ &+ (-a_3b_2m_3\dot{q}_3 - a_2a_3m_3\dot{q}_3)\sin(q_3) + (-a_3b_1m_3\dot{q}_2\\ &- a_3b_1m_3\dot{q}_3 - a_1a_3m_3\dot{q}_2 - a_1a_3m_3\dot{q}_3)\sin(q_2 + q_3)\end{aligned}$$

$$\begin{aligned}C_{34} &= (-a_1b_0m_1\dot{q}_0 - a_1b_0m_1\dot{q}_1 - a_1b_0m_2\dot{q}_0 - a_1b_0m_2\dot{q}_1\\ &- a_1b_0m_3\dot{q}_0 - a_1b_0m_3\dot{q}_1 - b_0b_1m_2\dot{q}_0 - b_0b_1m_2\dot{q}_1 - b_0b_1m_3\dot{q}_0\\ &- b_0b_1m_3\dot{q}_1)\cos(\phi - q_1) + (-a_2b_0m_2\dot{q}_0 - a_2b_0m_2\dot{q}_1\\ &- a_2b_0m_3\dot{q}_0 - a_2b_0m_2\dot{q}_2 - a_2b_0m_3\dot{q}_1 - a_2b_0m_3\dot{q}_2 - b_0b_2m_3\dot{q}_0\\ &- b_0b_2m_3\dot{q}_1 - b_0b_2m_3\dot{q}_2)\cos(q_1 - \phi + q_2) + (-a_3b_0m_3\dot{q}_0\\ &- a_3b_0m_3\dot{q}_1 - a_3b_0m_3\dot{q}_2 - a_3b_0m_3\dot{q}_3)\cos(q_1 - \phi + q_2 + q_3)\\ &+ (-a_2b_1m_2\dot{q}_2 - a_1b_2m_3\dot{q}_2 - a_2b_1m_3\dot{q}_2 - b_1b_2m_3\dot{q}_2 - a_1a_2m_2\dot{q}_2\\ &- a_1a_2m_3\dot{q}_2)\sin(q_2) + (-a_3b_2m_3\dot{q}_3 - a_2a_3m_3\dot{q}_3)\sin(q_3)\\ &+ (-a_3b_1m_3\dot{q}_2 - a_3b_1m_3\dot{q}_3 - a_1a_3m_3\dot{q}_2 - a_1a_3m_3\dot{q}_3)\sin(q_2 + q_3)\end{aligned}$$

$$\begin{aligned}C_{35} &= (-a_2b_0m_2\dot{q}_0 - a_2b_0m_2\dot{q}_1 - a_2b_0m_3\dot{q}_0 - a_2b_0m_2\dot{q}_2\\ &- a_2b_0m_3\dot{q}_1 - a_2b_0m_3\dot{q}_2 - b_0b_2m_3\dot{q}_0 - b_0b_2m_3\dot{q}_1 - b_0b_2m_3\dot{q}_2)\\ &\times \cos(q_1 - \phi + q_2) + (-a_3b_0m_3\dot{q}_0 - a_3b_0m_3\dot{q}_1 - a_3b_0m_3\dot{q}_2\\ &- a_3b_0m_3\dot{q}_3)\cos(q_1 - \phi + q_2 + q_3) + (-a_2b_1m_2\dot{q}_0 - a_1b_2m_3\dot{q}_0\\ &- a_2b_1m_2\dot{q}_1 - a_2b_1m_3\dot{q}_0 - a_1b_2m_3\dot{q}_1 - a_2b_1m_2\dot{q}_2 - a_2b_1m_3\dot{q}_1\\ &- a_1b_2m_3\dot{q}_2 - a_2b_1m_3\dot{q}_2 - b_1b_2m_3\dot{q}_0)\sin(q_2) + (-b_1b_2m_3\dot{q}_1\\ &- b_1b_2m_3\dot{q}_2 - a_1a_2m_2\dot{q}_0 - a_1a_2m_2\dot{q}_1 - a_1a_2m_3\dot{q}_0 - a_1a_2m_2\dot{q}_2\\ &- a_1a_2m_3\dot{q}_1 - a_1a_2m_3\dot{q}_2)\sin(q_2) + (-a_3b_2m_3\dot{q}_3 - a_2a_3m_3\dot{q}_3)\\ &\times \sin(q_3) + (-a_3b_1m_3\dot{q}_0 - a_3b_1m_3\dot{q}_1 - a_3b_1m_3\dot{q}_2 - a_3b_1m_3\dot{q}_3\\ &- a_1a_3m_3\dot{q}_0 - a_1a_3m_3\dot{q}_1 - a_1a_3m_3\dot{q}_2 - a_1a_3m_3\dot{q}_3)\sin(q_2 + q_3)\end{aligned}$$

$$\begin{aligned}C_{36} &= (-a_3b_0m_3(\dot{q}_0 + \dot{q}_1 + \dot{q}_2 + \dot{q}_3))\cos(q_1 - \phi + q_2 + q_3)\\ &+ (-a_3m_3(a_2 + b_2)(\dot{q}_0 + \dot{q}_1 + \dot{q}_2 + \dot{q}_3))\sin(q_3)\\ &+ (-a_3m_3(a_1 + b_1)(\dot{q}_0 + \dot{q}_1 + \dot{q}_2 + \dot{q}_3))\sin(q_2 + q_3)\end{aligned}$$

$$C_{41} = 0$$

$$C_{42} = 0$$

$$\begin{aligned}C_{43} &= (a_1b_0m_1\dot{q}_0 + a_1b_0m_2\dot{q}_0 + a_1b_0m_3\dot{q}_0 + b_0b_1m_2\dot{q}_0\\ &+ b_0b_1m_3\dot{q}_0)\cos(\phi - q_1) + (a_2b_0m_2\dot{q}_0 + a_2b_0m_3\dot{q}_0\\ &+ b_0b_2m_3\dot{q}_0)\cos(q_1 - \phi + q_2) + (a_3b_0m_3\dot{q}_0)\cos(q_1 - \phi + q_2 + q_3)\\ &+ (-a_2b_1m_2\dot{q}_2 - a_1b_2m_3\dot{q}_2 - a_2b_1m_3\dot{q}_2 - b_1b_2m_3\dot{q}_2 - a_1a_2m_2\dot{q}_2\\ &- a_1a_2m_3\dot{q}_2)\sin(q_2) + (-a_3b_2m_3\dot{q}_3 - a_2a_3m_3\dot{q}_3)\sin(q_3)\\ &+ (-a_3b_{1q}m_3\dot{q}_2 - a_3b_1m_3\dot{q}_3 - a_1a_3m_3\dot{q}_2 - a_1a_3m_3\dot{q}_3)\sin(q_2 + q_3)\end{aligned}$$

$$\begin{aligned}C_{44} &= (-a_2b_1m_2\dot{q}_2 - a_1b_2m_3\dot{q}_2 - a_2b_1m_3\dot{q}_2 - b_1b_2m_3\dot{q}_2\\ &- a_1a_2m_2\dot{q}_2 - a_1a_2m_3\dot{q}_2)\sin(q_2) + (-a_3b_2m_3\dot{q}_3 - a_2a_3m_3\dot{q}_3)\\ &\times \sin(q_3) + (-a_3b_1m_3\dot{q}_2 - a_3b_1m_3\dot{q}_3 - a_1a_3m_3\dot{q}_2 - a_1a_3m_3\dot{q}_3)\\ &\times \sin(q_2 + q_3)\end{aligned}$$

$$\begin{aligned}C_{45} &= (-a_2b_1m_2\dot{q}_0 - a_1b_2m_3\dot{q}_0 - a_2b_1m_2\dot{q}_1 - a_2b_1m_3\dot{q}_0\\ &- a_1b_2m_3\dot{q}_1 - a_2b_1m_2\dot{q}_2 - a_2b_1m_3\dot{q}_1 - a_1b_2m_3\dot{q}_2 - a_2b_1m_3\dot{q}_2\\ &- b_1b_2m_3\dot{q}_0)\sin(q_2) + (-b_1b_2m_3\dot{q}_1 - b_1b_2m_3\dot{q}_2 - a_1a_2m_2\dot{q}_0\\ &- a_1a_2m_2\dot{q}_1 - a_1a_2m_3\dot{q}_0 - a_1a_2m_2\dot{q}_2 - a_1a_2m_3\dot{q}_1 - a_1a_2m_3\dot{q}_2)\\ &\times \sin(q_2) + (-a_3b_2m_3\dot{q}_3 - a_2a_3m_3\dot{q}_3)\sin(q_3) + (-a_3b_1m_3\dot{q}_0\\ &- a_3b_1m_3\dot{q}_1 - a_3b_1m_3\dot{q}_2 - a_3b_1m_3\dot{q}_3 - a_1a_3m_3\dot{q}_0 - a_1a_3m_3\dot{q}_1\\ &- a_1a_3m_3\dot{q}_2 - a_1a_3m_3\dot{q}_3)\sin(q_2 + q_3)\end{aligned}$$

$$\begin{aligned}C_{46} &= (-a_3m_3(a_2 + b_2)(\dot{q}_0 + \dot{q}_1 + \dot{q}_2 + \dot{q}_3))\sin(q_3)\\ &+ (-a_3m_3(a_1 + b_1)(\dot{q}_0 + \dot{q}_1 + \dot{q}_2 + \dot{q}_3))\sin(q_2 + q_3)\end{aligned}$$

$$C_{51} = 0$$

$$C_{52} = 0$$

$$\begin{aligned}C_{53} &= (a_2b_0m_2\dot{q}_0 + a_2b_0m_3\dot{q}_0 + b_0b_2m_3\dot{q}_0)\cos(q_1 - \phi + q_2)\\ &+ (a_3b_0m_3\dot{q}_0)\cos(q_1 - \phi + q_2 + q_3) + (a_2b_1m_2\dot{q}_0 + a_1b_2m_3\dot{q}_0\\ &+ a_2b_1m_2\dot{q}_1 + a_2b_1m_3\dot{q}_0 + a_1b_2m_3\dot{q}_1 + a_2b_1m_3\dot{q}_1 + b_1b_2m_3\dot{q}_0\\ &+ b_1b_2m_3\dot{q}_1 + a_1a_2m_2\dot{q}_0 + a_1a_2m_2\dot{q}_1 + a_1a_2m_3\dot{q}_0\\ &+ a_1a_2m_3\dot{q}_1)\sin(q_2) + (-a_3b_2m_3\dot{q}_3 - a_2a_3m_3\dot{q}_3)\sin(q_3)\\ &+ (a_3b_1m_3\dot{q}_0 + a_3b_1m_3\dot{q}_1 + a_1a_3m_3\dot{q}_0 + a_1a_3m_3\dot{q}_1)\sin(q_2 + q_3)\end{aligned}$$

$$C_{54} = (a_2 b_1 m_2 \dot{q}_0 + a_1 b_2 m_3 \dot{q}_0 + a_2 b_1 m_2 \dot{q}_1 + a_2 b_1 m_3 \dot{q}_0 + a_1 b_2 m_3 \dot{q}_1 + a_2 b_1 m_3 \dot{q}_1 + b_1 b_2 m_3 \dot{q}_0 + b_1 b_2 m_3 \dot{q}_1 + a_1 a_2 m_2 \dot{q}_0 + a_1 a_2 m_2 \dot{q}_1 + a_1 a_2 m_3 \dot{q}_0 + a_1 a_2 m_3 \dot{q}_1) \sin(q_2) + (-a_3 b_2 m_3 \dot{q}_3 - a_2 a_3 m_3 \dot{q}_3) \sin(q_3) + (a_3 b_1 m_3 \dot{q}_0 + a_3 b_1 m_3 \dot{q}_1 + a_1 a_3 m_3 \dot{q}_0 + a_1 a_3 m_3 \dot{q}_1) \sin(q_2 + q_3)$$

$$C_{55} = (-a_3 m_3 \dot{q}_3 (a_2 + b_2)) \sin(q_3)$$

$$C_{56} = (-a_3 m_3 (a_2 + b_2)(\dot{q}_0 + \dot{q}_1 + \dot{q}_2 + \dot{q}_3)) \sin(q_3)$$

$$C_{61} = 0$$

$$C_{62} = 0$$

$$C_{63} = (a_3 b_0 m_3 \dot{q}_0) \cos(q_1 - \phi + q_2 + q_3) + (a_3 m_3 (a_2 \dot{q}_0 + a_2 \dot{q}_1 + a_2 \dot{q}_2 + b_2 \dot{q}_0 + b_2 \dot{q}_1 + b_2 \dot{q}_2)) \sin(q_3) + (a_3 m_3 (a_1 \dot{q}_0 + a_1 \dot{q}_1 + b_1 \dot{q}_0 + b_1 \dot{q}_1)) \sin(q_2 + q_3)$$

$$C_{64} = (a_3 m_3 (a_2 \dot{q}_0 + a_2 \dot{q}_1 + a_2 \dot{q}_2 + b_2 \dot{q}_0 + b_2 \dot{q}_1 + b_2 \dot{q}_2)) \sin(q_3) + (a_3 m_3 (a_1 \dot{q}_0 + a_1 \dot{q}_1 + b_1 \dot{q}_0 + b_1 \dot{q}_1)) \sin(q_2 + q_3)$$

$$C_{65} = (a_3 m_3 (a_2 + b_2)(\dot{q}_0 + \dot{q}_1 + \dot{q}_2)) \sin(q_3)$$

$$C_{66} = 0$$

Acknowledgments

This research was financially supported in part by the Natural Science and Engineering Research Council of Canada under the Canada Graduate Scholarship CGS M-623748-2017 and the Ontario Graduate Scholarship.

References

[1] "Technical Report on Space Debris," United Nations TR 99-82839, New York, 1999.

[2] Kessler, D. J., Johnson, N. L., Liou, J. C., and Matney, M., "The Kessler Syndrome: Implications to Future Space Operations," Guidance and Control 2010, *Advances in the Astronautical Sciences*, Vol. 137, Univelt, Escondido, CA, 2010, pp. 47–62; also American Astronautical Soc. Paper 10-016, Springfield, VA, 2010.

[3] Kessler, D. J., and Cour-Palais, B. G., "Collision Frequency of Artificial Satellites: The Creation of a Debris Belt," *Journal of Geophysical Research*, Vol. 83, No. A6, 1978, pp. 2637–2646. doi:10.1029/JA083iA06p02637

[4] Dubowsky, S., and Torres, M. A., "Path Planning for Space Manipulators to Minimize Spacecraft Attitude Disturbances," *1991 IEEE International Conference on Robotics and Automation*, IEEE Publ., Piscataway, NJ, 1991, pp. 2522–2528, http://ieeexplore.ieee.org/abstract/document/132005/ [retrieved 2019].

[5] Torres, M. A., and Dubowsky, S., "Minimizing Spacecraft Attitude Disturbances in Space Manipulator Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 1010–1017, https://arc.aiaa.org/doi/pdfplus/10.2514/3.20936 [retrieved 2019].

[6] Agrawal, O. P., and Xu, Y., "Global Optimum Path Planning for a Redundant Space Robot," Carnegie–Mellon Univ. CMU-RI-TR-91-15, Pittsburgh, PA, 1991, http://www.dtic.mil/docs/citations/ADA255316 [retrieved 2019].

[7] Papadopoulos, E., and Abu-Abed, A., "Design and Motion Planning for a Zero-Reaction Manipulator," *1994 IEEE International Conference on Robotics and Automation*, IEEE Publ., Piscataway, NJ, 1994, pp. 1554–1559, http://ieeexplore.ieee.org/abstract/document/351367/ [retrieved 2019].

[8] Lampariello, R., Agrawal, S., and Hirzinger, G., "Optimal Motion Planning for Free-Flying Robots," *2003 IEEE International Conference on Robotics and Automation*, Vol. 3, IEEE Publ., Piscataway, NJ, 2003, pp. 3029–3035, http://ieeexplore.ieee.org/abstract/document/1242056/ [retrieved 2019].

[9] Aghili, F., "Optimal Control for Robotic Capturing and Passivation of a Tumbling Satellite with Unknown Dynamics," *AIAA Guidance, Navigation and Control Conference and Exhibit*, AIAA Paper 2008-7274, 2008, http://arc.aiaa.org/doi/10.2514/6.2008-7274 [retrieved 2019]. doi:10.2514/6.2008-7274

[10] Yoshida, K., Dimitrov, D., and Nakanishi, H., "On the Capture of Tumbling Satellites by a Space Robot," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE Publ., Piscataway, NJ, 2006, pp. 4127–4132, http://ieeexplore.ieee.org/abstract/document/4059057/ [retrieved 2019].

[11] Huang, P., and Xu, Y., "PSO-Based Time-Optimal Trajectory Planning for Space Robot with Dynamic Constraints," *2006 IEEE International Conference on Robotics and Biomimetics*, IEEE Publ., Piscataway, NJ, 2006, pp. 1402–1407.

[12] Oki, T., Nakanishi, H., and Yoshida, K., "Time-Optimal Manipulator Control of a Free-Floating Space Robot with Constraint on Reaction Torque," *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008*, IEEE Publ., Piscataway, NJ, 2008, pp. 2828–2833, http://ieeexplore.ieee.org/abstract/document/4650835/ [retrieved 2019].

[13] Cascio, J. A., "Optimal Path Planning for Multi-Arm, Multi-Link Robotic Manipulators," M.S. Thesis, Naval Postgraduate School, Monterey, CA, 2008, http://www.dtic.mil/docs/citations/ADA547109 [retrieved 2019].

[14] Liu, H.-D., Shi, Y., Liang, B., Xu, W.-F., and Wang, X.-Q., "An Optimal Trajectory Planning Method for Stabilization of Coupled Space Robotic System After Capturing," *Procedia Engineering*, Vol. 29, Elsevier, Philadelphia, PA, 2012, pp. 3117–3123, http://linkinghub.elsevier.com/retrieve/pii/S1877705812004614 [retrieved 2019]. doi:10.1016/j.proeng.2012.01.451

[15] Xu, S., Wang, H., Zhang, D., and Yang, B., "Adaptive Reactionless Motion Control for Free-Floating Space Manipulators with Uncertain Kinematics and Dynamics," *IFAC Proceedings Volumes*, Vol. 46, No. 20, 2013, pp. 646–653, http://linkinghub.elsevier.com/retrieve/pii/S147466701631610X [retrieved 2019]. doi:10.3182/20130902-3-CN-3020.00145

[16] Flores-Abad, A., Wei, Z., Ma, O., and Pham, K., "Optimal Control of Space Robots for Capturing a Tumbling Object with Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 2014–2017, http://arc.aiaa.org/doi/10.2514/1.G000003 [retrieved 2019]. doi:10.2514/1.G000003

[17] Crain, A., and Ulrich, S., "Nonlinear Optimal Trajectory Planning for Free-Floating Space Manipulators Using a Gauss Pseudospectral Method," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA Paper 2016-5272, 2016, http://arc.aiaa.org/doi/10.2514/6.2016-5272 [retrieved 2019]. doi:10.2514/6.2016-5272

[18] Rybus, T., Seweryn, K., and Sasiadek, J. Z., "Trajectory Optimization of Space Manipulator with Non-Zero Angular Momentum During Orbital Capture Maneuver," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2016-0885, 2016, http://arc.aiaa.org/doi/10.2514/6.2016-0885 [retrieved 2019]. doi:10.2514/6.2016-0885

[19] Virgili-Llop, J., Drew, J., Zappulla, R., and Romano, M., "Autonomous Capture of a Resident Space Object by a Spacecraft with a Robotic Manipulator: Analysis, Simulation and Experiments," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA Paper 2016-5269, 2016, http://arc.aiaa.org/doi/10.2514/6.2016-5269 [retrieved 2019]. doi:10.2514/6.2016-5269

[20] Virgili-Llop, J., Drew, J. V., Zappulla, R., and Romano, M., "Laboratory Experiments of Resident Space Object Capture by a Spacecraft–Manipulator System," *Aerospace Science and Technology*, Vol. 71, Dec. 2017, pp. 530–545, http://linkinghub.elsevier.com/retrieve/pii/S1270963816311932 [retrieved 2019]. doi:10.1016/j.ast.2017.09.043

[21] Zappulla, R., Virgili-Llop, J., Zagaris, C., Park, H., and Romano, M., "Dynamic Air-Bearing Hardware-in-the-Loop Testbed to Experimentally Evaluate Autonomous Spacecraft Proximity Maneuvers," *Journal of Spacecraft and Rockets*, Vol. 54, No. 4, 2017, pp. 825–839, https://arc.aiaa.org/doi/10.2514/1.A33769 [retrieved 2019]. doi:10.2514/1.A33769

[22] Misra, G., and Bai, X., "Task-Constrained Trajectory Planning of Free-Floating Space-Robotic Systems Using Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 11, 2017, pp. 2857–2870, https://arc.aiaa.org/doi/10.2514/1.G002405 [retrieved 2019]. doi:10.2514/1.G002405

[23] Misra, G., and Bai, X., "Optimal Path Planning for Free-Flying Space Manipulators via Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 11, 2017, pp. 3026–3033,

https://arc.aiaa.org/doi/10.2514/1.G002487 [retrieved 2019]. doi:10.2514/1.G002487

[24] Yoshida, K., Hashizume, K., and Abiko, S., "Zero Reaction Maneuver: Flight Validation with ETS-VII Space Robot and Extension to Kinematically Redundant Arm," *2001 IEEE International Conference on Robotics and Automation*, Vol. 1, IEEE Publ., Piscataway, NJ, 2001, pp. 441–446, http://ieeexplore.ieee.org/abstract/document/932590/ [retrieved 2019].

[25] Park, H., Zappulla, R., Zagaris, C., Virgili-Llop, J., and Romano, M., "Nonlinear Model Predictive Control for Spacecraft Rendezvous and Docking with a Rotating Target," *27th AAS/AIAA Space Flight Mechanics Meeting*, AAS Paper 17-496, Springfield, VA, 2017.

[26] Rybus, T., Seweryn, K., and Sasiadek, J. Z., "Control System for Free-Floating Space Manipulator Based on Nonlinear Model Predictive Control (NMPC)," *Journal of Intelligent and Robotic Systems*, Vol. 85, Nos. 3–4, March 2017, pp. 491–509. doi:10.1007/978-3-319-94517-0_2

[27] Ulrich, S., "Direct Adaptive Control Methodologies for Flexible-Joint Space Manipulators with Uncertainties and Modeling Errors," Ph.D. Thesis, Carleton Univ., Ottawa, ON, Canada, 2012, https://curve.carleton.ca/system/files/etd/e9ba6c7c-817b-427e-9216-7dc507b14117/etd_pdf/853a59c7ba7534e82b1050a3c75f4c4a/ulrich-directadaptivecontrolmethodologiesforflexiblejoint.pdf [retrieved 2019].

[28] Umetani, Y., and Yoshida, K., "Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 3, 1989, pp. 303–314, http://ieeexplore.ieee.org/abstract/document/34766/ [retrieved 2019].

[29] Craig, J., *Introduction to Robotics: Mechanics and Control*, Pearson, Upper Saddle River, NJ, 2005.

[30] Tucker, M., and Perreira, D., "Generalized Inverses for Robotic Manipulators," *Mechanism and Machine Theory*, Vol. 22, No. 6, 1987, pp. 507–514.

[31] Benson, D., "A Gauss Pseudospectral Transcription for Optimal Control," Ph.D. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, 2005, https://dspace.mit.edu/handle/1721.1/28919 [retrieved 2019].

[32] Huntington, G. T., "Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems," Ph.D. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, 2007, http://ssl.mit.edu/publications/theses/PhD-2007-HuntingtonGeoffrey.pdf [retrieved 2019].

[33] Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1435–1440, http://arc.aiaa.org/doi/10.2514/1.20478 [retrieved 2019]. doi:10.2514/1.20478

[34] Huntington, G. T., Benson, D., and Rao, A. V., "Optimal Configuration of Tetrahedral Spacecraft Formations," *The Journal of the Astronautical Sciences*, Vol. 55, No. 2, 2007, pp. 141–169. doi:10.1007/BF03256518

[35] Huntington, G. T., and Rao, A. V., "Optimal Reconfiguration of Spacecraft Formations Using the Gauss Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 689–698, http://arc.aiaa.org/doi/10.2514/1.31083 [retrieved 2019]. doi:10.2514/1.31083

[36] Ross, I. M., and Karpenko, M., "A Review of Pseudospectral Optimal Control: From Theory to Flight," *Annual Reviews in Control*, Vol. 36, No. 2, 2012, pp. 182–197, http://linkinghub.elsevier.com/retrieve/pii/S1367578812000375 [retrieved 2019]. doi:10.1016/j.arcontrol.2012.09.002

[37] Ross, I. M., *A Primer on Pontryagin's Principle in Optimal Control*, Collegiate Publ., San Fransisco, CA, 2015.

[38] Holmström, K., Göran, A. O., and Edvall, M. M., *User's Guide for TOMLAB/SOL*, TOMLAB Optimization Inc., 2008, https://tomopt.com/tomlab/download/manuals.php.

[39] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131, http://epubs.siam.org/doi/10.1137/S0036144504446096 [retrieved 2019]. doi:10.1137/S0036144504446096

[40] Gong, Q., Fahroo, F., and Ross, I. M., "Spectral Algorithm for Pseudospectral Methods in Optimal Control," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 460–471, http://arc.aiaa.org/doi/10.2514/1.32908 [retrieved 2019]. doi:10.2514/1.32908

[41] Ross, I. M., and Fahroo, F., "Pseudospectral Knotting Methods for Solving Nonsmooth Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, 2004, pp. 397–405, http://arc.aiaa.org/doi/10.2514/1.3426 [retrieved 2019]. doi:10.2514/1.3426

[42] Philip, E., and Elizabeth, W., "Sequential Quadratic Programming Methods," UCSD Dept. of Mathematics TR NA-10-03, San Diego, CA, 2010.

[43] Gill, P. E., Saunders, M. A., and Murray, W., "User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming," Univ. of California, 2008, https://tomopt.com/tomlab/download/manuals.php.

[44] Sniedovich, M., "Dynamic Programming and Principles of Optimality," *Journal of Mathematical Analysis and Applications*, Vol. 65, No. 3, 1978, pp. 586–606, https://doi.org/10.1016/0022-247X(78)90166-X [retrieved 2019].

[45] Hovell, K., and Ulrich, S., "Experimental Validation for Tethered Capture of Spinning Space Debris," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2017-1049, 2017, http://arc.aiaa.org/doi/10.2514/6.2017-1049 [retrieved 2019]. doi:10.2514/6.2017-1049

[46] Hovell, K., and Ulrich, S., "Postcapture Dynamics and Experimental Validation of Subtethered Space Debris," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 2, 2018, pp. 519–525, https://arc.aiaa.org/doi/10.2514/1.G003049 [retrieved 2019]. doi:10.2514/1.G003049

[47] Ulrich, S., and Hovell, K., "Iterative Learning Control of Spacecraft Proximity Operations Based on Confidence Level," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2017-1046, 2017, http://arc.aiaa.org/doi/10.2514/6.2017-1046 [retrieved 2019]. doi:10.2514/6.2017-1046