SPACE Conferences and Exposition
13 - 16 September 2016, Long Beach, California
AIAA/AAS Astrodynamics Specialist Conference

Nonlinear Optimal Trajectory Planning for Free-Floating Space Manipulators using a Gauss Pseudospectral Method

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

In this paper, an optimal trajectory guidance law is developed for a free-floating space manipulator through a combination of a Gaussian pseudospectral collocation scheme with a Non-Linear Problem solver. Specifically, the resulting nonlinear optimal guidance problem with path constraints is defined with the general pseudospectral optimal software and numerically solved by the sparse nonlinear optimizer solver. By utilizing this solver, the pseudospectral method simultaneously solves the entire trajectory over a small number of nodes based on the path constraints, initial conditions, and initial guesses provided by the user. Simulations demonstrate the performance for a two degree-of-freedom space manipulator, and the results suggest an improvement in terms of attitude displacements, compared to previously-published results.

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 de-commissioned satellites and depleted rocket engines, most of which are travelling at speeds of up to 28,000 kph. 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] 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 is related to the autonomous robotic capture of uncooperative targets. Specifically, the reaction disturbance torque applied to the servicer robotic spacecraft due to the physical motion of the manipulator may cause the destabilization of the servicer spacecraft or severe damage to the robotic arm.

There have been many researchers who have proposed solutions to the problem of optimal trajectory planning. Dubowsky and Torres[3] worked on preliminary path planning for space manipulators. Agrawal and Xu[4] proposed a global optimum path planning technique for redundant space manipulators. Papadopoulos and Abu-Abed[5] introduced a motion planning technique for a zero-reaction manipulator. Lapariello et al.[6] presented a time optimal motion planning method using criteria in the joint space. Aghili[7] designed an optimal controller to capture a tumbling satellite using an objective function that minimized the operation time and relative velocity between the robot tip and the target. Oki et al.[8] also proposed an optimal control method for capturing a tumbling satellite, however they focused primarily on minimizing the operational time for fast capturing.

According to research published by Angel et al.,[9] disturbances to the attitude of the base can be greatly reduced through nonlinear optimal guidance laws, which predict the optimal future capture time, as well as the optimal manipulator trajectory and debris state, such that the resulting impact or disturbance on the attitude of the base satellite is minimized. Similarly to Angel et al.,[9] an optimal trajectory guidance

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

law is developed in this paper through a combination of a Gaussian pseudospectral collocation scheme with a Non-Linear Problem (NLP) solver, Specifically, the resulting nonlinear optimal guidance problem with path constraints is defined with the general pseudospectral optimal software (GPOPS-I) and numerically solved by the sparse nonlinear optimizer (SNOPT) NLP solver. By utilizing the SNOPT solver, GPOPS-I simultaneously solves the entire trajectory over a small number of nodes based on the path constraints, initial conditions, and initial guesses provided by the user. Some previous application examples can be found in Refs. 13–18. In the context of this work, the use of GPOPS-I allowed for the comparison, in numerical simulations, of the resulting optimal guidance trajectories against the work of Angel et al.,[9] in which the TOMLAB optimization software was used.

This paper is organized as follows: Section II outlines the general kinematic and dynamic equations that describe the manipulator motion. Section III details the optimal trajectory planning technique used, and any associated equations. Finally, Section IV presents the 2-DOF free-floating manipulator simulation used to demonstrate the performance of the technique that is being validated.

II. Manipulator Modelling

This section first outlines the general kinematic and dynamic equations that describe the motion of the manipulator. These general equations will then be used to derive the case-specific equations used to simulate a two degree-of-freedom free-floating planar manipulator.

A. Forward Kinematics

The forward kinematics problem for a manipulator is concerned with determining the end-effector position and orientation as a function of each individual joint angle. These kinematic equations are presented in the following subsection.

1. *General Equations*

The position of the end-effector is geometrically written as:[10]

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

where $\mathbf{r}_e \in \mathbb{R}^{2\times 1}$ is the end-effector position in the inertial frame, $\mathbf{r}_0 \in \mathbb{R}^{2\times 1}$ is the position of the Center of Mass (CM) of the servicer in the inertial frame, $\mathbf{b}_0 \in \mathbb{R}^{2\times 1}$ is the distance between the CM of the servicer and the shoulder joint, and $\mathbf{L}_i \in \mathbb{R}^{2\times 1}$ is the length of each link in the robotic arm.

2. *Kinematics for a 2-DOF Free-Floating Manipulator*

The geometry of the 2-DOF free-floating free-floating manipulator simulated in Section IV is described in Fig. 1:

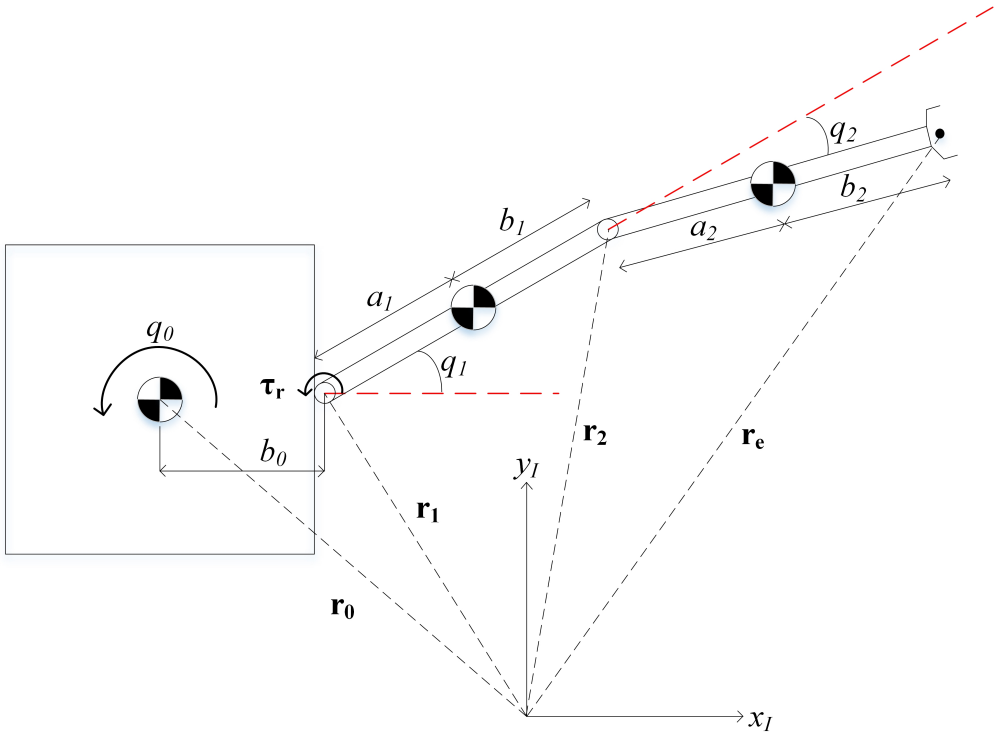


Figure 1. Example of a 2-DOF free-floating space manipulator

The kinematics at position-level for this system can be derived using Eq. (1). The resulting equations are given by:

$$x_e = x_0 + \|\mathbf{b}_0\| \cos(q_0) + \|\mathbf{L}_1\| \cos(q_0 + q_1) + \|\mathbf{L}_2\| \cos(q_0 + q_1 + q_2) \tag{2}$$

$$y_e = y_0 + \|\mathbf{b}_0\| \sin(q_0) + \|\mathbf{L}_1\| \sin(q_0 + q_1) + \|\mathbf{L}_2\| \sin(q_0 + q_1 + q_2) \tag{3}$$

where x_e and y_e are the components of end-effector position in the inertial frame, x_i and y_i are the position of each joint i in the inertial frame, and q_i is the joint angle for each link i.

The components of the end-effector velocity, $\dot{x}_e$ and $\dot{y}_e$, can be obtained by time differentiating Eqs. (2) and (3).

B. Dynamic Modelling

The following subsections outline the general derivation used to obtain the equations of motion for any free-floating manipulator.

1. *General Equations*

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 follows:[11]

$$L = T - U \tag{4}$$

where L is the Lagrangian, T is the kinetic energy component of the Lagrange equation, and U is the potential energy component of the Lagrangian.

The kinetic and potential energies are defined in terms of generalized coordinates $\mathbf{q} \in \mathbb{R}^n$, which encompass the position of the base satellite in inertial space as well as the joint angles. The equations of motion for the two-link planar free-floating manipulator subjected to joint torques is derived according to the following equation:

$$\tau_i = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \left(\frac{\partial L}{\partial q_i}\right); \forall i = 1, 2 \tag{5}$$

where τ_i is the generalized joint torque for joint i. The kinetic energy is assumed to be a quadratic function of the generalized coordinates $\mathbf{q}$ of the form:

$$T = \frac{1}{2}\sum_{i=0}^{n}\sum_{j=0}^{n} M_{ij}\dot{q}_i\dot{q}_j \tag{6}$$

where M_{ij} are the components of the robots inertia matrix, denoted $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n\times n}$, which is symmetric and positive definite. In the case of a free-floating space manipulator with rigid links, there is no source of potential energy. Therefore, the Lagrangian can be simplified to:

$$L = \frac{1}{2}\sum_{i=0}^{n}\sum_{j=0}^{n} M_{ij}\dot{q}_i\dot{q}_j \tag{7}$$

The partial derivative of the Lagrangian with respect to the kth joint position is given by:

$$\frac{\partial L}{\partial q_k} = \frac{1}{2}\sum_{i=0}^{n}\sum_{j=0}^{n} \frac{\partial M_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j \tag{8}$$

Similarly, the partial derivative of the Lagrangian with respect to the kth joint velocity is given by:

$$\frac{\partial L}{\partial q_k} = \sum_{j=0}^{n} M_{ij}\dot{q}_j \tag{9}$$

Therefore, the derivative of Eq. (9) with respect to time is given by:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}_k}\right) = \sum_{j=0}^{n} M_{ij}\ddot{q}_j + \sum_{i=0}^{n}\sum_{j=0}^{n} \frac{\partial M_{ij}}{\partial q_i}\dot{q}_i\dot{q}_j \tag{10}$$

Thus, for each joint, the Euler-Lagrange equations are given by:

$$\tau_k = \sum_{j=0}^{n} M_{ij}\ddot{q}_j + \sum_{i=0}^{n}\sum_{j=0}^{n}\left(\frac{\partial M_{ij}}{\partial q_i} - \frac{1}{2}\frac{\partial M_{ij}}{\partial q_k}\right)\dot{q}_i\dot{q}_j \tag{11}$$

It can be shown that:

$$\sum_{i=0}^{n}\sum_{j=0}^{n}\left(\frac{\partial M_{ij}}{\partial q_i} - \frac{1}{2}\frac{\partial M_{ij}}{\partial q_k}\right)\dot{q}_i\dot{q}_j = \sum_{i=0}^{n}\sum_{j=0}^{n} c_{ijk}\dot{q}_i\dot{q}_j \tag{12}$$

where the terms c_{ijk} are known as the Christoffel symbols. The Euler-Lagrange equations of motion can then be written as:

$$\tau_k = \sum_{j=0}^{n} M_{ij}\ddot{q}_j + \sum_{i=0}^{n}\sum_{j=0}^{n} c_{ijk}\dot{q}_i\dot{q}_j \tag{13}$$

where terms of the type q_i^2 are called centrifugal, while those of the type $\dot{q}_i\dot{q}_j$ are called the Coriolis terms. This expression can be rewritten once more into a more intuitive matrix form:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \tag{14}$$

where the k,jth element of the centrifugal/Coriolis matrix as denoted by $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{n\times n}$ is defined as:

$$C_{ij} = \sum_{i=1}^{n} c_{ijk}\dot{q}_i \tag{15}$$

For any n-link manipulator, the inertia matrix $\mathbf{M}(\mathbf{q})$ can be derived in terms of the linear and angular velocity components, which are functions of the Jacobian matrix and the derivatives of the joint variables. The Jacobian matrix is simply a more convenient way to formulate the velocity kinematics of the manipulator. Hence, the translational contribution to the total kinetic energy, $\mathbf{T}_v \in \mathbb{R}^{n\times n}$, is:

$$\mathbf{T}_v = \frac{1}{2}\sum_{i=0}^{n} m_i \mathbf{v}_i^2 = \frac{1}{2}\sum_{i=0}^{n} \dot{\mathbf{q}}^T \left(m_i \mathbf{J}_{cv_i}^T \mathbf{J}_{cv_i}\right)\dot{\mathbf{q}} \tag{16}$$

where m_i is the mass of body i, $\mathbf{v}_i$ is the translational velocity of joint i, and $\mathbf{J}_{cv_i}$ is the translational velocity Jacobian matrix at CM of body i. Similarly, the rotational contribution to the total kinetic energy, $\mathbf{T}_\omega \in \mathbb{R}^{n\times n}$, is:

$$\mathbf{T}_\omega = \frac{1}{2}\sum_{i=0}^{n} I_i \boldsymbol{\omega}_i^2 = \frac{1}{2}\sum_{i=0}^{n} \dot{\mathbf{q}}^T \left(I_i \mathbf{J}_{c\omega_i}^T \mathbf{J}_{c\omega_i}\right)\dot{\mathbf{q}} \tag{17}$$

where I_i is the inertia of body i, $\boldsymbol{\omega}_i$ is the rotational velocity of joint i, and $\mathbf{J}_{c\omega_i}$ is the rotational velocity Jacobian matrix at CM of body i. The inertia matrix $\mathbf{M}(\mathbf{q})$ is obtained by adding the two matrices defined in Eq. (16) and (17), as follows:

$$\mathbf{M}(\mathbf{q}) = \frac{1}{2}\left[\sum_{i=0}^{n} \dot{\mathbf{q}}^T \left(m_i \mathbf{J}_{cv_i}^T \mathbf{J}_{cv_i}\right)\dot{\mathbf{q}} + \sum_{i=0}^{n} \dot{\mathbf{q}}^T \left(I_i \mathbf{J}_{c\omega_i}^T \mathbf{J}_{c\omega_i}\right)\dot{\mathbf{q}}\right] \tag{18}$$

In this work, the planar dynamics were obtained symbolically using MATLAB.

2. *Velocity Kinematics - The Jacobian*

To facilitate the derivation of the dynamic equations, it is preferable to formulate the velocity kinematics in a more convenient differential form. This is accomplished by deriving a suitable Jacobian that relates the linear and angular velocity of the CM for each body, denoted by $\mathbf{v}_{c_i} \in \mathbb{R}^{2\times 1}$ and $\boldsymbol{\omega}_{c_i} \in \mathbb{R}^{3\times 1}$, to the joint velocity, as follows:

$$\mathbf{v}_{c_i} = \mathbf{J}_{cv_i}(\mathbf{q})\dot{\mathbf{q}} \tag{19}$$

$$\boldsymbol{\omega}_{c_i} = \mathbf{J}_{c\omega_i}(\mathbf{q})\dot{\mathbf{q}} \tag{20}$$

where by $\mathbf{J}_{cv_i}(\mathbf{q}) \in \mathbb{R}^{2\times 5}$, $\mathbf{J}_{cw_i}(\boldsymbol{q}) \in \mathbb{R}^{3\times 5}$. Equations (19) and (20) can be grouped together and represented by:

$$\begin{bmatrix} v_{cx_i} \\ v_{cy_i} \\ v_{cx_i} \\ \omega_{cx_i} \\ \omega_{cy_i} \\ \omega_{cz_i} \end{bmatrix} = \mathbf{J}_{c_i}(\mathbf{q}) \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \tag{21}$$

where $\mathbf{J}_{c_i}(\mathbf{q}) \in \mathbb{R}^{5\times 5}$ denotes the Jacobian matrix, which is given by the following expression for each joint:

$$\mathbf{J}_{c_0}(\mathbf{q}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{22}$$

$$\mathbf{J}_{c_1}(\mathbf{q}) = \begin{bmatrix} 1 & 0 & -b_0 S_0 - a_1 S_1 & -a_1 S_1 & 0 \\ 0 & 1 & b_0 C_0 + a_1 C_1 & a_1 C_1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{23}$$

$$\mathbf{J}_{c_2}(\mathbf{q}) = \begin{bmatrix} 1 & 0 & -b_0 S_0 - L_1 S_1 - a_2 S_2 & -L_1 S_1 - a_2 S_2 & -a_2 S_2 \\ 0 & 1 & b_0 C_0 + L_1 C_1 + a_2 C_2 & L_1 C_1 + a_2 C_2 & a_2 C_2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{24}$$

where:

$S_0 = \sin(q_0)$
$S_1 = \sin(q_0 + q_1)$
$S_2 = \sin(q_0 + q_1 + q_2)$
$C_0 = \cos(q_0)$
$C_1 = \cos(q_0 + q_1)$
$C_2 = \cos(q_0 + q_1 + q_2)$
$L_1 = a_1 + b_1$

3. Derived Dynamic Equations for a 2-DOF Planar Manipulator

Using the Jacobian matrices defined in Eqs. (22) − (24), the translational contribution to the total kinetic energy can be written as:

$$\mathbf{T}_v = \frac{1}{2}\dot{\mathbf{q}}^T (m_0 \mathbf{J}_{cv_0}^T \mathbf{J}_{cv_0} + m_1 \mathbf{J}_{cv_1}^T \mathbf{J}_{cv_1} + m_2 \mathbf{J}_{cv_2}^T \mathbf{J}_{cv_2}) \dot{\mathbf{q}} \tag{25}$$

Similarly, the rotational contribution to the total kinetic energy can be written as:

$$\mathbf{T}_\omega = \frac{1}{2}\dot{\mathbf{q}}^T (I_0 \mathbf{J}_{c\omega_0}^T \mathbf{J}_{c\omega_0} + I_1 \mathbf{J}_{c\omega_1}^T \mathbf{J}_{c\omega_1} + I_2 \mathbf{J}_{c\omega_2}^T \mathbf{J}_{c\omega_2}) \dot{\mathbf{q}} \tag{26}$$

As defined in Eq. (18), the inertia matrix $\mathbf{M}(\mathbf{q})$ is obtained by adding the two matrices in Eq. (25) and (26). The resulting matrix is the 5×5 positive definite matrix as shown:

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} \\ M_{31} & M_{32} & M_{33} & M_{34} & M_{35} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} \end{bmatrix} \tag{27}$$

where $M_{ij}\ \forall i = 1...5, j = 1...5$ are defined in Appendix A. Using the components of the inertia matrix, the Christoffel symbols can be derived using Eq. (12). Then, from the Christoffel symbols with Eq. (15), the expression for the 5×5 centrifugal/Coriolis matrix can be derived:

$$\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} \end{bmatrix} \tag{28}$$

where $C_{ij}\ \forall i = 1...5, j = 1...5$ are also defined in Appendix A.

III. Optimal Trajectory Planning

This section describes the optimal trajectory planning method used. This includes the optimization software used, as well as the NLP solver. Finally, this section formulates the state space form of the dynamic equations, as required by the optimization software.

A. Approach Overview

The optimal trajectory problem was defined and solved within the MATLAB-Simulink environment using GPOPS-I and the Sparse Nonlinear Optimizer, SNOPT. The overview presented by Kedare[12] will be described in the following sections. While a full description of GPOPS-I is beyond the scope of this paper, the interested readers can find detailed formulations in Benson,[13] Huntington,[14] Benson et al.,[15,16] and Huntington et al.[17,18] The initial and final manipulator states, as well as the controller bounding box required by GPOPS-I were obtained from the problem defined by Angel et al.[9]

B. GPOPS-I

GPOPS-I was implemented alongside the dynamics model presented in Section II.B to obtain the results, as well as validate the capability of the MATLAB-Simulink environment to accurately optimize the manipulator trajectory. Following the validation, GPOPS-I was utilized for the generation of the optimal trajectory. The software, which is integrated into MATLAB, requires the user to define the dynamics of the problem using differential equations, an associated cost function, connections between phases (if any), and finally the limits and initial guesses for each of the states and controls.

1. *GPOPS Algorithm*

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

$$J = \Phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t)\mathrm{d}t \tag{29}$$

where Φ is the Mayer, t_i is the initial time, t_f is the final time, and g is the Lagrange component. The system is subject to the dynamic constraints given by:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{30}$$

and the associated boundary conditions:

$$\phi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) = 0 \tag{31}$$

For this research, the optimal guidance problem defined in Eqs. (29) and (30) is solved using a direct transcription method called the *Gauss Pseudospectral Method*. It is important to note that the cost function and constraints that define the 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.

C. Sparse Nonlinear Optimizer (SNOPT)

The NLP described in Section IV was solved with the MATLAB *mex* interface of the NLP solver SNOPT using analytic first-order derivatives for the contained Jacobian and the gradient of the objective function. A large number of solvers, such as IPOPT, KNITRO, and MINOS, are available to academic and industrial users. However, SNOPT was selected based on availability and prior implementation in the GPOPS environment.

SNOPT 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. SNOPT uses a sequential quadratic programming (SQP) algorithm to solve an NLP. The search direction is obtained through QP 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 implementation of SNOPT may be found in the SNOPT Users Guide.[19] For the research presented in this paper, it is sufficient to note that SNOPT is suitable for solving nonlinear problems of the form:

$$\text{minimize } \mathbf{x} \text{ for } f_0(\mathbf{x}) \text{ subject to } l \leq \begin{pmatrix} \mathbf{x} \\ f(\mathbf{x}) \\ A_L\mathbf{x} \end{pmatrix} \leq u$$

where $\mathbf{x}$ is an n-vector of variables, l and u are constant lower and upper bounds, $f_0(\mathbf{x})$ is a smooth scalar objective function, A_L is a sparse matrix, and $f(\mathbf{x})$ is a vector of smooth nonlinear constraint functions $f_i(\mathbf{x})$. In the ideal case, the first derivative of $f_0(\mathbf{x})$ and $f_i(\mathbf{x})$ should be coded by the user. Upper and lower bounds are specified for all variables and constraints.

D. State-Space Manipulator Dynamics and Cost Function

GPOPS-I requires that the dynamic equations presented in Eq. (14) be rewritten into a state space-form, as follows:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\boldsymbol{\tau} \tag{32}$$

where $\mathbf{x} \in \mathbb{R}^{2n}$ is the state vector; $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^{2n}$ is nonlinear function of the states; $\mathbf{G}(\mathbf{x}) \in \mathbb{R}^{2n\times n}$ is the control matrix; and $\boldsymbol{\tau} \in \mathbb{R}^n$ are the joint torques. These can be defined as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \tag{33}$$

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}^{5\times5} & \mathbf{1}^{5\times5} \\ \mathbf{0}^{5\times5} & -\mathbf{M}(\mathbf{x_1})^{-1}\mathbf{C}(\mathbf{x_1},\mathbf{x_2}) \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} \tag{34}$$

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}^{5\times5} \\ \mathbf{M}(\mathbf{x_1})^{-1} \end{bmatrix} \tag{35}$$

Similar to Angel et al.,[9] the assumption in this paper is made that the base satellite can rotate and translate freely. Thus, it is possible to determine the impact of the manipulators motion on the attitude of the servicing satellite by deriving the reaction torque on the shoulder (base) of the manipulator arm. This reaction torque can be defined by:

$$\boldsymbol{\tau}_r = -\sum_{i=1}^{n} \left(I_i\boldsymbol{\alpha}_i + (\mathbf{r}_i - \mathbf{b}_0)^{\times}(m_i\mathbf{v}_i)\right) \tag{36}$$

where $\boldsymbol{\tau}_r \in \mathbb{R}^n$ is the moment that the manipulator applies at its shoulder. The objective of this optimal control problem is therefore to find a time history for each joints control torque such that, when the end-effector of the manipulator is controlled by this set of torques to move from its initial to its final pose, it will have minimal impact on the attitude of the base satellite. To find this set of optimal control torques, we can define the following objective function:

$$\mathbf{J} = \int_0^{t_f} \left(\boldsymbol{\tau}_r^T\boldsymbol{\tau}_r\right) \mathrm{d}t \tag{37}$$

IV. Simulation Example

To validate the results obtained in Ref. 9, the same simulation example will be used. The primary difference between the two simulations is the use of the GPOPS-I optimization software instead of the TOMLAB optimization software. However, both simulations make use of the NLP solver SNOPT. The

simulation example presented in this section uses the 2-DOF planar manipulator in Fig. 1, whose parameters are defined in Table 1.

Table 1. Parameters for the 2-DOF Free-Floating Manipulator.

Body	Body Number	a_i (m)	b_i (m)	m_i (kg)	I_i (kg $\cdot$ m^2)
Base Satellite	0	—	0.5	1100	183
Link #1	1	0.5	0.5	25	2.5
Link #2	2	0.5	0.5	25	2.5

A. Nominal Optimal Control Setup

The optimal control problem can be defined as follows:

Minimize:

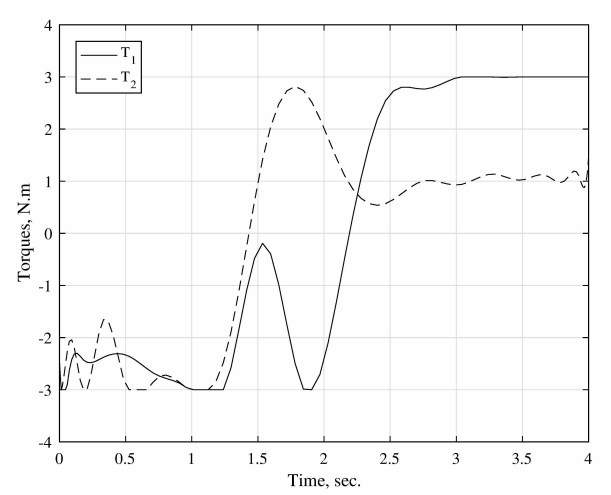
$$\mathbf{J}(\mathbf{x}, \dot{\mathbf{x}}, \boldsymbol{\tau}) = \int_0^{t_f} \boldsymbol{\tau}_r^T \boldsymbol{\tau}_r \mathrm{d}t$$

Subject to:

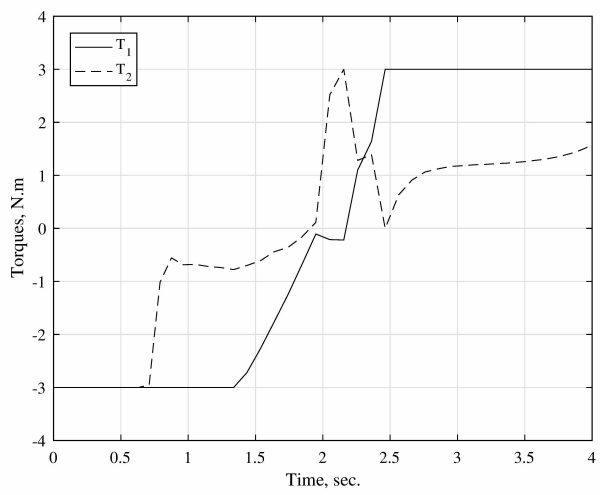
- Dynamic Constraints:
  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\boldsymbol{\tau}$
- Kinematic Boundaries:
  $q_1(t_0) = 0.53$ rad
  $q_2(t_0) = 0.778$ rad
  $q_1(t_f) = 0.615$ rad
  $q_2(t_f) = -0.326$ rad
- Final Time:
  $t_f = 4$ s
- Box Constraints:
  -3 N $\cdot$ m $\leq \tau_1 \leq 3$ N $\cdot$ m, -3 N $\cdot$ m $\leq \tau_2 \leq 3$ N $\cdot$ m
  $-\frac{\pi}{2}$ rad $\leq q_1 \leq \frac{\pi}{2}$ rad, $-\frac{\pi}{2}$ rad $\leq q_2 \leq \frac{\pi}{2}$ rad

B. Simulation Results

Using the control setup described in the previous section, simulations were performed to determine the effectiveness of the optimal trajectory guidance law. The results obtained for the simulations are presented below. The optimal control torques obtained are shown in Fig. 2.



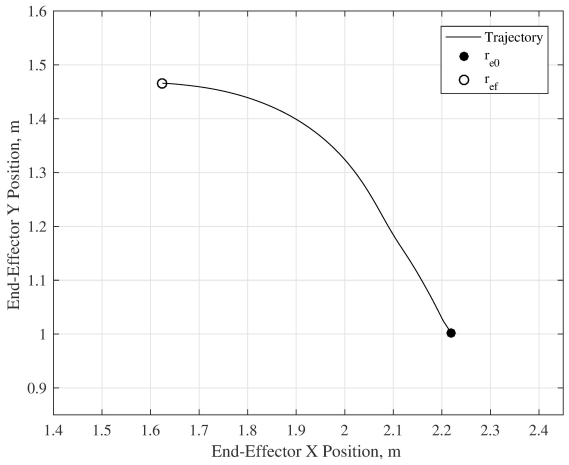
(a) Joint control torques (GPOPS-I).



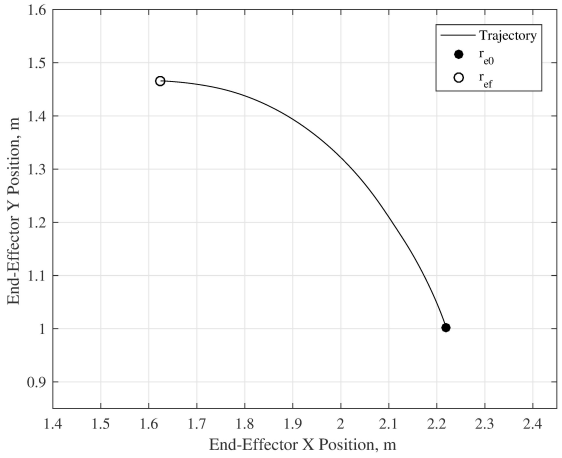
(b) Joint control torques (TOMLAB).[9]

Figure 2. Optimal control torques obtained using GPOPS and TOMLAB

While these control torques do differ slightly, the resulting final position of the end-effector is still the same. The end-effector trajectory obtained for both GPOPS-I and TOMLAB are presented in Fig. 3.



(a) End-Effector Trajectory (GPOPS-I).



(b) End-Effector Trajectory (TOMLAB).[9]

Figure 3. Optimal control torques obtained using GPOPS and TOMLAB

As expected, the result obtained from GPOPS-I are comparable to those obtained using TOMLAB. In both cases, the reaction torque $\boldsymbol{\tau}_r$ produced by the manipulator on the base satellite has been minimized, as shown in Fig. 4.

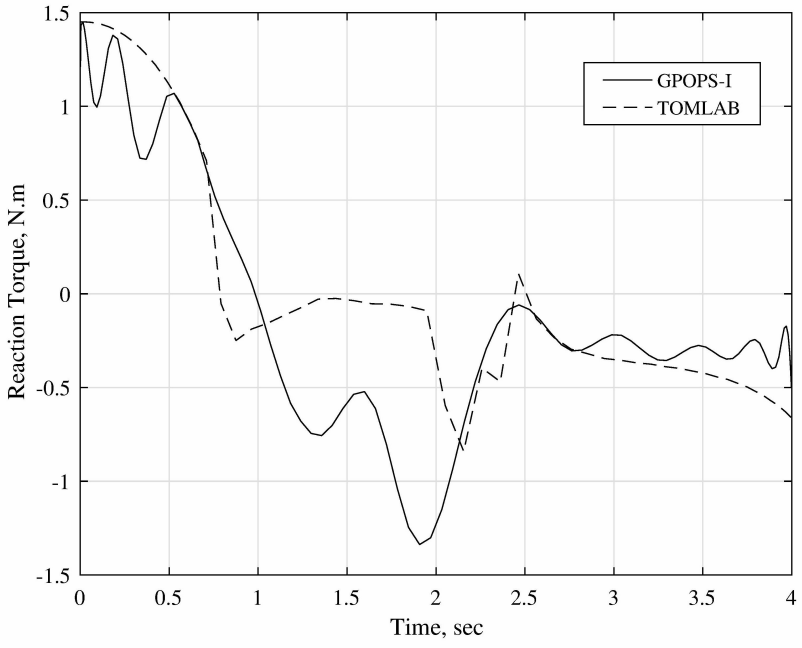


Figure 4. Reaction torque at the base of the manipulator for GPOPS-I and TOMLAB[9]

Compared to the results obtained by Angel et al.[9] using the TOMLAB optimization tool, numerical simulations indicate that the results obtained with the proposed GPOPS-I method are more effective in minimizing the attitude displacement of the servicer spacecraft during a manipulator reconfiguration maneuver, as shown in Fig. 5. This improvement is attributable to the different optimization tools used.

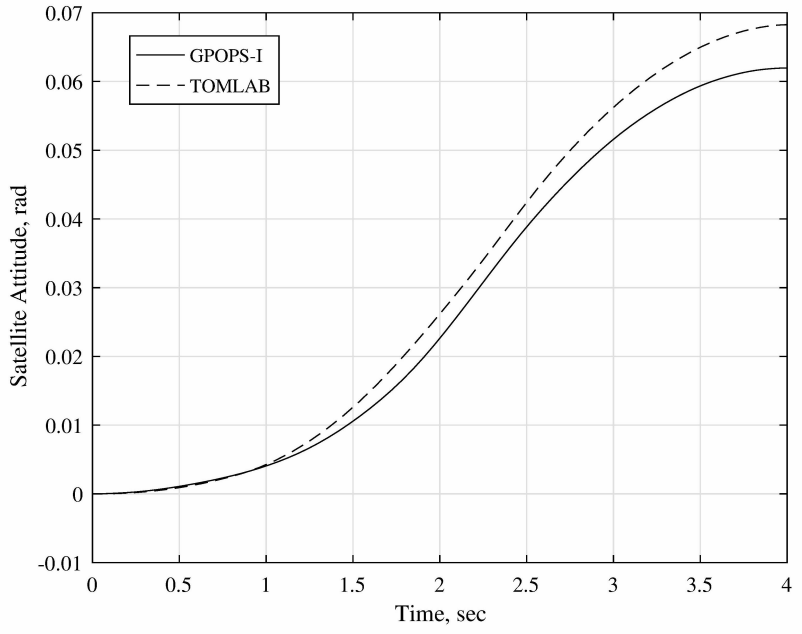


Figure 5. Attitude displacement predicted for GPOPS-I and TOMLAB

A closed-loop transpose Jacobian non-optimal feedback controller was then used to verify that the attitude disturbance had indeed been minimized. The resulting comparison is shown in Fig. 6.

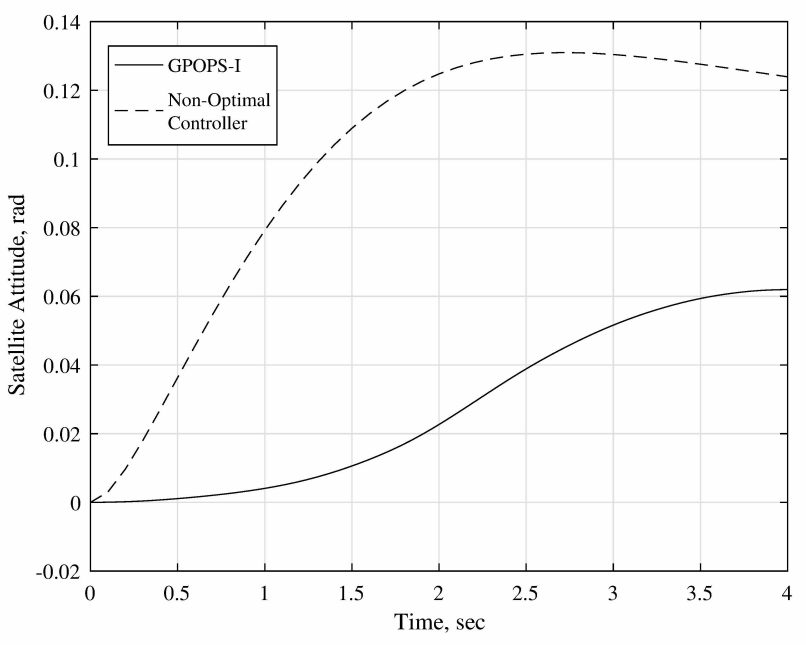


Figure 6. Attitude displacement predicted by GPOPS-I and a non-optimal controller

Clearly, the attitude displacement has been significantly minimized from the case where no optimization was done. This can be further validated by plotting the reaction torque for the non-optimal controller, as shown in Fig. 7.

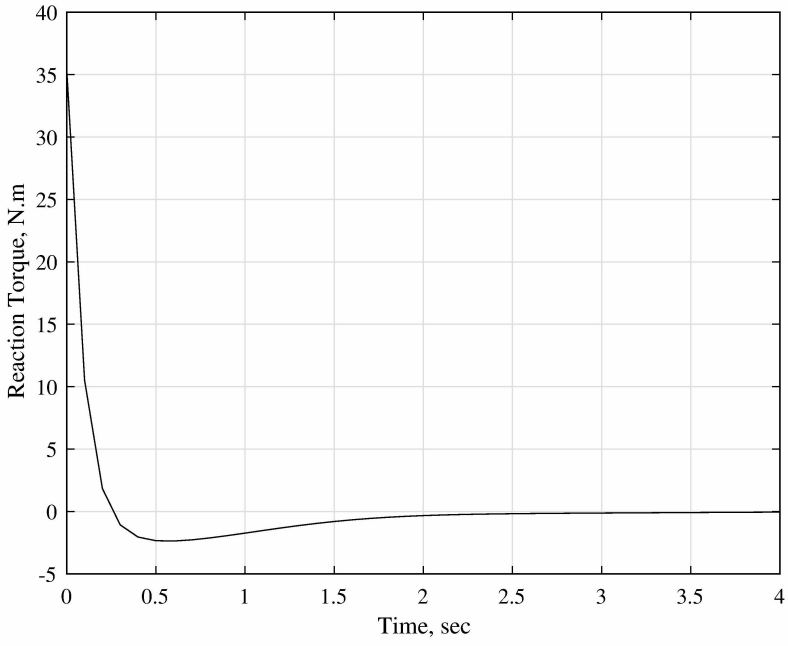


Figure 7. Reaction torque for the non-optimal controller

The reaction torque for the non-optimal case is significantly larger then it was for the optimized case. This indicates that the technique validated in this paper is very effective at reducing the attitude displacement resulting from the motion of the manipulator.

V. Conclusion

In this paper, a nonlinear optimal guidance law for free-floating manipulators was validated. This was accomplished by first deriving the kinematic and dynamic equations for a planar free-floating manipulator, which described the motion of the end-effector as a function of the generalized joint coordinates. Next, the optimal trajectory planning technique was presented and a cost function was defined. Finally, a dynamic simulation for a 2-DOF free-floating manipulator was used to demonstrate the performance of the technique, and the results compared favourably to past research. Ultimately, this paper demonstrated that the nonlinear optimal guidance law is more effective at minimizing the attitude disturbance of a base satellite, compared to an existing optimal guidance law and to a non-optimal transpose Jacobian control law. Future work will include the experimental validations of the proposed optimization scheme.

A. Components of the Inertia and Coriolis Matrices

The components $M_{ij}\ \forall i = 1...5, j = 1...5$ are presented below:

$$
\begin{aligned}
M_{11} &= m_0 + m_1 + m_2 \\
M_{12} &= 0 \\
M_{13} &= m_1(-b_0S_0 - a_1S_1) + m_2(-b_0S_0 - L_1S_1 - a_2S_2) \\
M_{14} &= -m_1S_1a_1 + m_2(-L_1S_1 - a_2S_2) \\
M_{15} &= -a_2m_2S_2 \\
M_{21} &= 0 \\
M_{22} &= m_0 + m_1 + m_2 \\
M_{23} &= m_1(b_0C_0 + a_1C_1) + m_2(b_0C_0 + L_1C_1 + a_2C_2) \\
M_{24} &= m_1C_1a_1 + m_2(L_1C_1 + a_2C_2) \\
M_{25} &= a_2m_2C_2 \\
M_{31} &= M_{13} \\
M_{32} &= M_{23} \\
M_{33} &= m_1(-b_0S_0 - a_1S_1)^2 + m_1(b_0C_0 + a_1C_1)^2 + m_2(-b_0S_0 - L_1S_1 - a_2S_2)^2 \\
&\quad + m_2(b_0C_0 + L_1C_1 + a_2C_2)^2 + I_0 + I_1 + I_2 \\
M_{34} &= -m_1(-b_0S_0 - a_1S_1)(a_1S_1) + m_1(b_0C_0 + a_1C_1)(a_1C_1) + m_2 \\
&\quad (-b_0S_0 - L_1S_1 - a_2S_2)(-L_1S_1 - a_2S_2) + m_2(b_0C_0 + L_1C_1 + a_2C_2)(L_1C_1 + a_2C_2) + I_1 + I_2 \\
M_{35} &= -m_2(-b_0S_0 - L_1S_1 - a_2S_2)(a_2S_2) + m_2(b_0 + L_1C_1 + a_2C_2)(a_2C_2) + I_2 \\
M_{41} &= M_{14} \\
M_{42} &= M_{24} \\
M_{43} &= M_{34} \\
M_{44} &= m_1(a_1S_1)^2 + m_1(a_1C_1)^2 + m_2(-L_1S_1 - a_2S_2)^2 + m_2(L_1C_1 + a_2S_2)^2 + m_2(L_1C_1 + a_2C_2)^2 + I_1 + I_2 \\
M_{45} &= -m_2(-L_1S_1 - a_2S_2)(a_2S_2) + m_2(L_1C_1 + a_2C_2)(a_2C_2) + I_2 \\
M_{51} &= M_{15} \\
M_{52} &= M_{25} \\
M_{53} &= M_{35} \\
M_{54} &= M_{45} \\
M_{55} &= m_2(a_2S_2)^2 + m_2(a_2C_2)^2 + I_2
\end{aligned}
$$

where:

$b_0 = \|\mathbf{b}_0\|$
$L_1 = \|\mathbf{L}_1\|$
$L_2 = \|\mathbf{L}_2\|$

From these components, as described in the paper, the components $C_{ij}\ \forall i = 1...5, j = 1...5$ are derived and presented below:

$$
\begin{aligned}
C_{11} &= 0 \\
C_{12} &= 0 \\
C_{13} &= \dot{q}_0(m_1(-C_0b_0 - C_1a_1) + m_2(-C_0b_0 - C_1L_1 - C_2a_2)) + \dot{q}_1(-m_1C_1a_1 + m_2(-C_1L_1 - C_2a_2)) - \\
&\quad \dot{q}_2m_2C_2a_2 \\
C_{14} &= \dot{q}_0(-m_1C_1a_1 + m_2(-C_1L_1 - C_2a_2)) + \dot{q}_1(-m_1C_1a_1 + m_2(-C_1L_1 - C_2a_2)) - \dot{q}_2m_2C_2a_2) \\
C_{15} &= -\dot{q}_0m_2a_2C_2 - \dot{q}_1m_2a_2C_2 - \dot{q}_2m_2a_2C_2 \\
C_{21} &= 0 \\
C_{22} &= 0 \\
C_{23} &= \dot{q}_0(m_1(-S_0b_0 - S_1a_1) + m_2(-S_0b_0 - S_1L_1 - S_2a_2)) + \dot{q}_1(-m_1S_1a_1 + m_2(-S_1L_1 - S_2a_2)) - \dot{q}_2m_2S_2a_2 \\
C_{24} &= \dot{q}_0(-m_1S_1a_1 + m_2(-S_1L_1 - S_2a_2)) + \dot{q}_1(-m_1S_1a_1 + m_2(-S_1L_1 - S_2a_2)) - \dot{q}_2m_2S_2a_2 \\
C_{25} &= -\dot{q}_0m_2a_2S_2 - \dot{q}_1m_2a_2S_2 - \dot{q}_2m_2a_2S_2 \\
C_{31} &= 0 \\
C_{32} &= 0
\end{aligned}
$$

$$
\begin{aligned}
C_{33} &= \dot{q}_0(m_1(-S_0b_0 - S_1a_1)(-C_0b_0 - C_1a_1) + m_1(C_0b_0 + C_1a_1)(-S_0b_0 - S_1a_1) + \\
&\quad m_2(-S_0b_0 - S_1L_1 - S_2a_2)(-C_0b_0 - C_1L_1 - C_2a_2) + m_2(C_0b_0 + C_1L_1 + C_2a_2)(-S_0b_0 - S_1L_1 - S_2a_2)) \\
&\quad + \dot{q}_1(-m_1(-S_0b_0 - S_1a_1)C_1a_1 - m_1(C_0b_0 + C_1a_1)S_1a_1 + m_2(-S_0b_0 - S_1L_1 - S_2a_2)(-C_1L_1 - C_2a_2) \\
&\quad + m_2(C_0b_0 + C_1L_1 + C_2a_2)(-S_1L_1 - S_2a_2)) + \dot{q}_2(-m_2(-S_0b_0 - S_1L_1 - S_2a_2)C_2a_2 \\
&\quad - m_2(C_0b_0 + C_1L_1 + C_2a_2)S_2a_2) \\
C_{34} &= \dot{q}_0(-m_1(-S_0b_0 - S_1a_1)C_1a_1 - m_1(C_0b_0 + C_1a_1)S_1a_1 \\
&\quad + m_2(-S_0b_0 - S_1L_1 - S_2a_2)(-S_0b_0 - S_1a_1) + m_2(C_0b_0 + C_1L_1 + C_2a_2) \\
&\quad (-S_1L_1 - S_2a_2)) + \dot{q}_1(-m_1(-S_0b_0 - S_1a_1)C_1a_1 - m_1(C_0b_0 + C_1a_1)S_1a_1 \\
&\quad + m_2(-S_0b_0 - S_1L_1 - S_2a_2)(-C_1L_1 - C_2a_2) + m_2(C_0b_0 + C_1L_1 + C_2a_2) \\
&\quad (-S_1L_1 - S_2a_2)) + \dot{q}_2(-m_2(-S_0b_0 - S_1L_1 - S_2a_2)C_2a_2 - m_2(C_0b_0 + C_1L_1 + C_2a_2)S_2a_2) \\
C_{35} &= \dot{q}_0(-m_2(-S_0b_0 - S_1L_1 - S_2a_2)C_2a_2 - m_2(C_0b_0 + C_1L_1 + C_2a_2)S_2a_2) \\
&\quad + \dot{q}_1(-m_2(-S_0b_0 - S_1L_1 - S_2a_2)C_2a_2 - m_2(C_0b_0 + C_1L_1 + C_2a_2)S_2a_2) \\
&\quad + \dot{q}_2(-m_2(-S_0b_0 - S_1L_1 - S_2a_2)C_2a_2 - m_2(C_0b_0 + C_1L_1 + C_2a_2)S_2a_2) \\
C_{41} &= 0 \\
C_{42} &= 0 \\
C_{43} &= \dot{q}_0(-m_1(-C_0b_0 - C_1a_1)S_1a_1 + m_2(-C_0b_0 - C_1L_1 - C_2a_2)(-S_1L_1 - S_2a_2) \\
&\quad + m_2(-S_0b_0 - S_1L_1 - S_2a_2)(C_1L_1 + C_2a_2) + m_1(-S_0b_0 - S_1a_1)C_1a_1) \\
&\quad + \dot{q}_1(m_2(-C_1L_1 - C_2a_2)(-S_1L_1 - S_2a_2) + m_2(-S_1L_1 - S_2a_2)(C_1L_1 + C_2a_2)) \\
&\quad + \dot{q}_2(-m_2S_2a_2(C_1L_1 + C_2a_2) - m_2(-S_1L_1 - S_2a_2)C_2a_2) \\
C_{44} &= \dot{q}_0(m_2(-C_1L_1 - C_2a_2)(-S_1L_1 - S_2a_2) + m_2(-S_1L_1 - S_2a_2)(C_1L_1 + C_2a_2)) \\
&\quad + \dot{q}_1(m_2(-C_1L_1 - C_2a_2)(-S_1L_1 - S_2a_2) + m_2(-S_1L_1 - S_2a_2)(C_1L_1 + C_2a_2)) \\
&\quad + \dot{q}_2(-m_2S_2a_2(C_1L_1 + C_2a_2) - m_2(-S_1L_1 - S_2a_2)C_2a_2) \\
C_{45} &= \dot{q}_0(-m_2S_2a_2(C_1L_1 + C_2a_2) - m_2(-S_1L_1 - S_2a_2)C_2a_2) \\
&\quad + \dot{q}_1(-m_2S_2a_2(C_1L_1 + C_2a_2) - m_2(-S_1L_1 - S_2a_2)C_2a_2) \\
&\quad + \dot{q}_2(-m_2S_2a_2(C_1L_1 + C_2a_2) - m_2(-S_1L_1 - S_2a_2)C_2a_2) \\
C_{51} &= 0 \\
C_{52} &= 0 \\
C_{53} &= \dot{q}_0(-m_2(-C_0b_0 - C_1L_1 - C_2a_2)S_2a_2 + m_2(-S_0b_0 - S_1L_1 - S_2a_2)C_2a_2) \\
&\quad + \dot{q}_1(-m_2(-C_1L_1 - C_2a_2)S_2a_2 + m_2(-S_1L_1 - S_2a_2)C_2a_2) \\
C_{54} &= \dot{q}_0(-m_2(-C_1L_1 - C_2a_2)S_2a_2 + m_2(-S_1L_1 - S_2a_2)C_2a_2) \\
&\quad + \dot{q}_1(-m_2(-C_1L_1 - C_2a_2)S_2a_2 + m_2(-S_1L_1 - S_2a_2)C_2a_2) \\
C_{55} &= 0
\end{aligned}
$$

Acknowledgments

The authors would like to express their sincere gratitude to Angel Flores-Abad from the New Mexico State University for providing the raw data that helped us benchmark the new results obtained in this work.

References

[1]Kessler, D. J., Johnson, N. L., Liou, J.-C., and Matney, M., "The Kessler Syndrome: Implications to Future Space Operations," *in 33rd Annual AAS Guidance and Control Conference*, San Diego, CA, 2010, Feb. 6–10.

[2]"Enabling Large-Body Active Debris Removal Project." Space Technology Mission Directorate (STMD), NASA, TX, United States, 2015.

[3]Dubowsky, S. and Torres, M., "Path Planning for Space Manipulators to Minimize Spacecraft Attitude Disturbances," *in IEEE International Conference on Robotics and Automation, Sacramento*, CA., 1991, pp. 2522-2528.

[4]Agrawal, O. P. and Xu, Y., "On the global optimum path planning for redundant space manipulators," *in IEEE Transactions on Man and Cybernetics*, 1994, pp. 1306–1316.

[5]Papadopoulos, E. Abu-Abed, A., "Design and Motion Planning for a Zero-Reaction Manipulator," *in IEEE International Conference on Robotics and Automation*, San Diego, CA., 1994, pp. 1554–1559.

[6]Lampariello, R., Agrawal, S. and Hirzinger, H., "Optimal Motion Planning for Free-Flying Robots," *in IEEE International Conference on Robotics and Automation*, Taipei, Taiwan., 2003, pp. 3029-3035.

[7]Aghili, F., "Optimal Control for Robotic Capturing and Passivation of a Tumbling Satellite with Unknown Dynamics" *in AIAA Guidance Navigation and Control Conference*, Honolulu, Hawaii, 2008, pp. 1–21.

[8]Oki, T. Nakanishi, H and Yoshida, K., "Time-Optimal Manipulator Control of a Free-Floating Space Robot with Constraint on Reaction Torque" *in IEEE International Conference on Intelligent Robots and Systems*, Nice, France, 2008, pp. 2828-2833.

[9]Flores-Abad, A. Wei, A. Ma, O. 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.

[10]Yoshida, K. and Umetani, Y., *Space Robotics: Dynamics and Control*. Tokyo: Kluwer Academic Publishers, 1993.

[11]Ulrich, S., "Direct Adaptive Control Methodologies for Flexible-Joint Space Manipulators with Uncertainties and Modelling Errors." Ph.D. thesis, Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, Canada, 2012.

[12]Kedare, S., "Space Environment Modelling and Torque-Optimal Guidance for CubeSat Applications." M.A.Sc. thesis, Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, Canada, 2014.

[13]Benson, D. A., "A Gauss Pseudospectral Transcription for Optimal Control", Ph.D. Thesis, Dept of Aeronautics and Astronautics, MIT, November 2004.

[14]Huntington, G. T., "Advancements and Analysis of Gauss Pseudospectral Transcription for Optimal Control", Ph.D. Thesis, Dept of Aeronautics and Astronautics, MIT, May 2007.

[15]Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optmization and Costate Estimation via an Orthagonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, 2006, pp. 1435–1440.

[16]Benson, D. A., Huntington, G. T., and Rao, A. V., "Design of Optimal Tetrahedral Spacecraft Formations," Journal of Astronautical Sciences, vol. 55, no. 2, 2007, pp. 141–169.

[17]Huntington, G. T., Benson, D. A., How, J. P., Kanizay, N., Darby, C. L., and Rao, A. V., "Computation of Boundary Controls Using a Gauss Pseudospectral Method," *in 2007 Astrodynamics Specialist Conference*, Mackinac Island, Michigan, August 2007.

[18]Huntington, G. T., and Rao, A. V., "Optimal Reconfiguration of Spacecraft Formations Using a Gauss Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 3, 2008, pp. 689–698.

[19]Gill, P. E. Murray, W. and Saunders, M. A., User Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming, June 2008.

[20]Espero, M. T., "Future Space Robotics and Large Optical Systems: A Picture of Orbital Express," Ares V Workshop, California, USA, 2008.