

Trajectory Generation for Flexible-Joint Space Manipulators

David S. Carabis^{1,*} and John T. Wen²

¹Rensselaer Polytechnic Institute, Department of Mechanical, Aerospace, and Nuclear Engineering, Troy, New York, USA

²Rensselaer Polytechnic Institute, Department of Electrical, Computer, and Systems Engineering, Troy, New York, USA

Abstract

Space manipulator arms often exhibit significant joint flexibility and limited motor torque. Future space missions, including satellite servicing and large structure assembly, may involve the manipulation of massive objects, which will accentuate these limitations. Currently, astronauts use visual feedback on-orbit to mitigate oscillations and trajectory following issues. Large time delays between orbit and Earth make ground teleoperation difficult in these conditions, so more autonomous operations must be considered to remove the astronaut resource requirement and expand robotic capabilities in space. Trajectory planning for autonomous systems must therefore be considered to prevent poor trajectory tracking performance. We provide a model-based trajectory generation methodology that incorporates constraints on joint speed, motor torque, and base actuation for flexible-joint space manipulators while minimizing total trajectory time. Full spatial computer simulation results, as well as physical experiment results with a single-joint robot on an air bearing table, show the efficacy of our methodology.

Keywords:

space robotics, satellite servicing, trajectory planning, trajectory optimization, object manipulation, floating base

1 Introduction

Robotic manipulation is an attractive technology for many on-orbit applications. These applications include satellite servicing, space structure construction, and debris management (Flores-Abad et al., 2014; Tasker and Henshaw, 2008; Yoshida, 2003; Moosavian and Papadopoulos, 2007; Ogilvie et al., 2008). The use of autonomous robots for these tasks (with on-ground instruction) can help to reduce the need for astronaut resources, as well as pave the way for widespread adoption by industry and governmental space agencies alike.

Space manipulators are often lightweight, meaning they have limited motor torques and are flexible. This type of design is often necessary to meet payload requirements (Yamano et al., 2000). Link flexibility may arise from long lightweight links, while joint flexibility may arise from large gear ratios and harmonic drives (Flores-Abad et al., 2014). However, joint flexibility due to harmonic drives often has lower resonant modes than link flexibility, and is generally the dominant source of flexibility when considering captured object transport (Flores-Abad et al., 2014; Alberts et al., 1992). One example is the shuttle Remote Manipulator System (Canadarm), where joint flexibility generates the first brakes-on natural frequency at 0.03 Hz, while the lowest resonant mode due to link flexibility is at about 23 Hz (note that this data is for the typical payload grasping configuration, with grasped payload) (Alberts et al., 1992).

Many objects that would be manipulated on-orbit are also massive, such as a satellite in a servicing application. One example is the Hubble Space Telescope, which has a mass of 12,000 (Garner, 2018). Servicing applications are illustrated in Figure 1. The combination of limited motor torque, joint flexibility, and massive payload can lead to accuracy reductions, excessive vibrations, and trajectory following difficulties (Flores-Abad et al., 2014). Dangerous scenarios, such as object collision with the manipulator base or self-collision, may arise.

Regardless of these limitations, negligible gravitational effects allow such arms to transport massive objects. Two flight examples of lightweight arms are Canadarm and Canadarm 2 (also known as the shuttle and International Space Station (ISS) arms) (Scott et al., 1993; Ma et al., 1997). However, these arms are often teleoperated by astronauts during large object transport, such as when the SpaceX Dragon was docked at the ISS (Kauderer, 2012). Visual feedback can then be used by astronauts to avoid oscillatory behavior and significant trajectory following error. This capability is not available for ground teleoperation scenarios, where significant visual time delay is present (Flores-Abad et al., 2014).

In spite of these challenges, successful on-orbit capture and transportation tasks have taken place using limited ground control. Capture and docking of a cooperative satellite has been demonstrated by NASA’s ETS-VII satellite (Yoshida, 2003). Autonomous capture has been demonstrated by the DARPA Orbital Express Demonstration System (Flores-Abad et al., 2014). However, a key aspect of these two operations is a relatively small target object mass compared to the robot base (400 kg to 2,500 kg for ETS-VII (Kasai et al., 1999); 226 kg to 952 kg for the Orbital Express Demonstration System (Bell, 2020)). Both operations also involved cooperative target satellites (i.e., satellites designed for manipulation).

Most recently, Northrop Grumman’s MEV-1 satellite successfully docked with a non-cooperative satellite to extend lifetime after the target satellite’s fuel was depleted. MEV-1 rigidly docked with the target satellite and now acts as a propulsion stage for both satellites, rather than refueling and restoring propulsion (Northrop Grumman Space Logistics, 2020; Chang, 2020). This differs from robotic docking and servicing in several key ways.

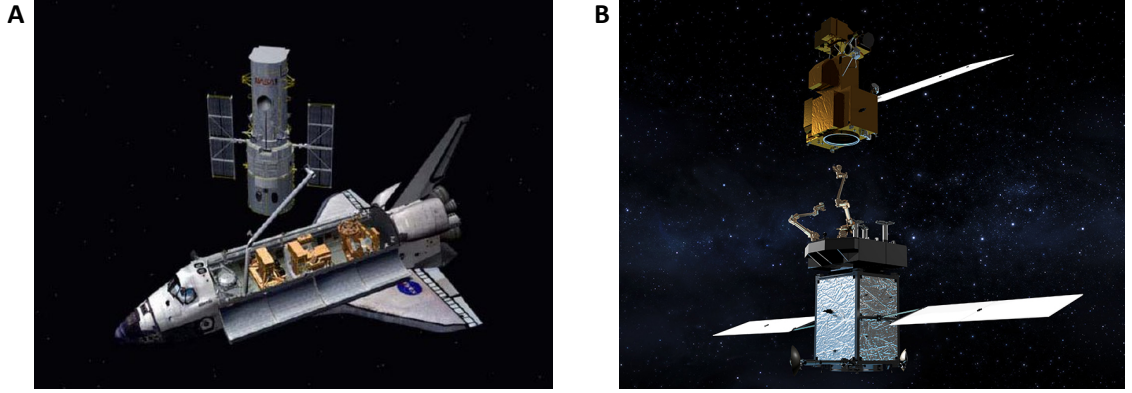


Figure 1: Examples of lightweight articulated manipulators for massive object capture and transport: **(A)** the Hubble telescope during a repair mission, **(B)** satellite servicer capturing a satellite. Photo credit: NASA.

First, MEV-1 utilizes an apogee kick thruster cone for docking (Northrop Grumman Space Logistics, 2020). This feature is present on approximately 80% of satellites in geosynchronous orbit (Hall, 2019), meaning that there are satellites that cannot be serviced via this method. Furthermore, this feature may be less prevalent in satellites in low Earth orbit. The fact that MEV-1 acts as a propulsion stage also means that only one target satellite can be helped at a time, as opposed to a refueling operation where both satellites must only remain docked for the duration of the refueling procedure. The presence of a robotic manipulator also opens the door to other servicing and repair procedures, such as replacing satellite components (Flores-Abad et al., 2014).

This does of course beg the question of why cone docking and a robotic arm are not used in tandem. While this is a potential solution for some scenarios, there are certain satellites that would not be able to be serviced, as stated above. This setup would also require additional launch mass for a servicing satellite. Cone docking also does not address the manipulation of other massive objects in space, such as components of space structures and scientific samples from asteroids.

Even with these successful on-orbit operations, robotic manipulation in space remains an open research area due to challenges with the manipulation of massive non-cooperative objects (here defined as objects not designed for grasping and manipulation) (Flores-Abad et al., 2014). Modeling of flexible joint space manipulators has been addressed. Some existing models have simplified the system by assuming the motor kinetic energy is only due to relative motion about the motor axis (Spong, 1987; Yu and Chen, 2014), while other models have incorporated the inertial coupling of the motors and arm links (Nanos and Papadopoulos, 2015; Ulrich et al., 2012). Furthermore, (Ulrich et al., 2012) has incorporated nonlinear spring and friction models in the flexible joints. Models

with both flexible joints and flexible links have also been investigated (Yu and Chen, 2014).

Flexible joint control has also been addressed (Spong, 1987; Yin et al., 2018; Sun et al., 2018; Spong, 1989; De Luca et al., 1989), including for space manipulation with a floating base. Control utilizing an additional link-side torque sensor and an inner torque feedback loop has been applied to a single flexible joint in simulation (Ferretti et al., 2005). The limitation of needing an additional link-side torque sensor was later addressed by (Ulrich and Sasiadek, 2012a), where an extended Kalman filter was used to estimate the link-side state for a 2 degree-of-freedom (DOF) robot in simulation. Adaptive control has been utilized to reduce trajectory tracking error for a 2 DOF robot with a floating base (Ulrich et al., 2012; Ulrich and Sasiadek, 2012b,a). However, this strategy leads to large motor torque values when there are sharp changes in the end-effector path. Computed torque (Yu and Chen, 2014) and feedback linearization (Nanos and Papadopoulos, 2015) have both been proposed, but both methods rely on model knowledge and are often not robust to modeling error. Singular perturbation theory has also been popular for designing flexible joint controllers, and has been applied to a dual-arm 2 DOF space robot scenario (Zarafshan and Moosavian, 2011), as well as a 2 DOF space robot with an additional flexible link (Yu and Chen, 2014). Outer-loop (i.e., link-side joint speed) control for flexible-joint manipulation of massive objects has been addressed, as well as path planning and checking planned trajectories against dynamic constraints, for fixed-base scenarios (Carabis et al., 2021). However, trajectory optimization and the influence of floating-base dynamics were not addressed in this work.

Trajectory generation for rigid space manipulators has been addressed using particle swarm optimization (Huang and Xu, 2006) and by extending the well-known minimum-time solution developed by (Bobrow et al., 1985) to floating base scenarios (Dubowsky et al., 1989). Trajectory generation for flexible-link space robot scenarios has also been addressed by (Wu et al., 2004), where oscillations during transport are minimized. However, this work does not address motor torque limits or joint flexibility, and some oscillations still persist even in simulation.

Our contribution is a model-based methodology for flexible-joint space manipulator trajectory planning. Existing control methods for flexible-joint manipulators do not account for joint speed and motor torque limits, and there will be issues with trajectory following in scenarios where these limits are reached. Our work incorporates constraints on motor torque and joint speed during the trajectory generation step, providing a methodology to plan trajectories where these constraints are satisfied. We examine both free-floating and controlled base scenarios, and incorporate base actuation (provided through reaction wheels and thrusters) constraints in the controlled base scenario. An optimization step is undertaken to reduce total trajectory time.

Furthermore, we investigate a 7 DOF space robot model with flexible joints and a free-flying base in simulation. Existing work has addressed relatively simple, low-DOF scenarios in simulation. Finally, we generate a feedforward motor

signal while planning the link-side trajectory. Although disturbances and modeling errors will require feedback control on-application, we propose the feedforward motor signal as a means to further reduce oscillations during trajectory execution. We investigate this experimentally using a single flexible-joint experiment conducted on an air-bearing table, a popular method for offsetting the effects of gravity (Rybus and Seweryn, 2016).

This paper is organized as follows. Section 2 provides the problem statement and scope, Section 3 provides the problem formulation, while Section 4 details the trajectory generation algorithm and optimization problem. Sections 5 and 6 provide the experimental setup and results, respectively. Finally, Section 7 offers a concluding discussion and outlines potential future work. Portions of this paper have also appeared in the doctoral dissertation Carabis (2020).

2 Problem Statement and Scope

We assume an n DOF flexible-joint space robot with only rotational joints, a floating base, and a grasped load. Define $q \in \mathbb{R}^n$ as the vector of link-side joint angles. Assume a desired joint-space path has been planned, which also defines the desired task-space path. Let $\lambda \in [0, 1]$ be the path variable, such that $q(\lambda = 0)$ defines the initial joint configuration and $q(\lambda = 1)$ defines the final joint configuration. The trajectory generation step then becomes indexing the path variable with time t , i.e. generating $\lambda(t)$.

The problem under consideration is generating the trajectory $\lambda(t)$ subject to motor torque, link-side joint speed, and base actuation constraints (in the controlled base scenario). Mathematically stated, we require

$$|\dot{q}(t)| \preceq \dot{q}_{\max}, \quad |\tau_m(t)| \preceq \tau_{m,\max}, \quad \forall t, \quad (1)$$

for all scenarios considered, and

$$|f_b(t)| \preceq f_{b,\max}, \quad \forall t, \quad (2)$$

for scenarios with a controlled base. Here, $\preceq$ denotes an element-wise inequality, $\dot{q}_{\max}$ is the link-side joint speed limit, $\tau_m \in \mathbb{R}^n$ is the applied motor torque, $\tau_{m,\max}$ is the motor torque limit, f_b is the spatial force applied at the base, and $f_{b,\max}$ is the base spatial force limit. Note that the spatial force is stacked torque and force, i.e., $f_b = [\tau_b^T \ F_b^T]^T$, where τ_b is the torque generated by thrusters or reaction wheels and F_b is the force generated by thrusters at the robot base.

We note that there are additional dynamic limits that may be of concern, such as the speed of the grasped object at the end effector and spatial forces at the end effector leading to slip or gripper damage. The former can be incorporated into constraints on $\dot{q}$, while we consider the latter to be outside the scope of this work.

As there are many trajectories that would satisfy these requirements (and arbitrarily slowing a trajectory would make the constraints easier to satisfy), we

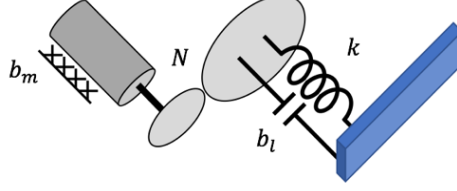


Figure 2: Single flexible joint. In this diagram, b_m , b_l , N , and k represent the motor damping, gear damping, gear ratio, and spring constant, respectively, and i denotes the i th joint.

propose finding a trajectory subject to the constraints while minimizing total trajectory time. Global minimum-time trajectory generation for this scenario remains an unsolved problem, and we allow for local minimum solutions. Furthermore, we allow for a sufficiently smooth trajectory shape to be assumed. Future work will seek to remove this assumption, potentially allowing for faster trajectories to be generated. Regardless, the methodology presented here provides a means for generating feasible trajectories and reducing the total trajectory time while satisfying dynamic constraints.

3 Problem Formulation

We model the flexible joints using the same approach as (Spong, 1987). This model assumes that each flexible joint is comprised of a 1-D rotational motor inertia and a torsional spring-damper that connects the i th motor to the i th manipulator link. Furthermore, this model assumes that the motors rotate relative to an inertial frame (i.e., the inertial coupling between the links and motors is ignored). Please note that we have derived a model using the Newton-Euler approach (Jain, 2010) that includes the inertial coupling between the motors and links, but no significant error was noted in simulation. This is likely due to the large mass/inertia of the manipulator base and grasped object, and we therefore utilize the model with no inertial coupling. Figure 2 illustrates this model for a single joint.

Using the spatial algebra approach (Jain, 2010) the dynamics of the manipulator arm, floating base, and grasped object are

$$\begin{bmatrix} f_b \\ \tau \end{bmatrix} = \begin{bmatrix} M_b & M_{rb} \\ M_{br} & M_r \end{bmatrix} \begin{bmatrix} \alpha_B \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} C_B \\ C_r \end{bmatrix} + \begin{bmatrix} \Phi_{n+1,B}^T \\ J^T \end{bmatrix} f. \quad (3)$$

Appendix A provides a detailed derivation. Here, τ denotes the link-side torque vector. The 6×6 mass-inertia matrix of the robot base is given by M_b , M_r is the joint-space mass-inertia matrix of the manipulator arm, M_{rb} , M_{br} are the mass-inertia couplings between the manipulator base and arm, C_r , C_B are the centrifugal and Coriolis terms, J is the well-known manipulator Jacobian,

and $\Phi_{n+1,B}^T$ is a mapping from the external spatial force f (generated by the grasped object motion) to the manipulator base center-of-mass (COM) frame B . Note that $\Phi_{n+1,B}^T$ is dependent on q , and that $n+1$ denotes the end effector frame. Further details on this mapping are provided in Appendix A. The spatial acceleration of the base is given by $\alpha_B = [\dot{\omega}_B^T \dot{v}_B^T]^T$, where ω_B is the rotational velocity of the manipulator base frame and v_B is the translational velocity of the manipulator base frame.

In the case where the floating base is perfectly controlled and acts as a fixed base, equation (3) becomes the well-known manipulator model with joint flexibility given by (Spong, 1987). When the base is uncontrolled and free-floating, $f_b = 0$.

We assume the target object is rigidly grasped, and model the object dynamics using

$$A^T f = M_c \alpha_c + b_c, \quad (4)$$

where A is the Jacobian transform between the end-effector and object COM, M_c is the object mass-inertia matrix (about the COM), b_c are the centrifugal/Coriolis terms associated with the grasped object motion, and c denotes the object COM frame. Since there is a single rigid grasp point, A^T is invertible and we can solve for f . As we consider a rigid grasp scenario in this paper, the mass/inertia of the grasped payload can be combined with the end effector link (i.e., set $f = 0$ and combine mass/inertia properties of link n and the payload). However, the full model presented here allows for contact models to be introduced in future work, so we present the arm and payload dynamics in full.

The joint torque for each flexible joint is given by

$$\tau = K(N^{-1}q_m - q) + b_l(N^{-1}\dot{q}_m - \dot{q}), \quad (5)$$

where $q_m \in \mathbb{R}^n$ is the vector of motor positions, K is a diagonal matrix of joint spring stiffnesses, b_l is a diagonal matrix of joint damping, and N a diagonal matrix of gear ratios. Note that we assume linear torsional springs and viscous damping. The motor dynamics are

$$I_m \ddot{q}_m + b_m \dot{q}_m + K N^{-1}(N^{-1}q_m - q) + b_l N^{-1}(N^{-1}\dot{q}_m - \dot{q}) = \tau_m, \quad (6)$$

where I_m , b_m are diagonal matrices of the motor inertias and viscous motor damping, respectively.

4 Trajectory Generation Algorithm

Our first step is to generate a feedforward motor trajectory that will result in some desired link-side trajectory, $q_d(t)$. We assume perfect motor control and consider the input to the system to be the motor trajectory, i.e. $u(t) = q_m(t)$. First, we consider a fixed-base system and assume $q_d(t)$ is provided. We combine the arm and load dynamics, as well as the portion of the joint torque dependent on link-side trajectory, into the term

$$v(t) = M_r \ddot{q} + C_r + K q + b_l \dot{q} + J^T f. \quad (7)$$

Note that the load dynamics $f(t)$ can be solved for in terms of $q_d(t)$ using velocity and acceleration propagation from the fixed-base. From equation (3), we can solve for $u(t) = q_m(t)$ using the low-pass filter (LPF)

$$u_i(s) = \frac{v_i(s)}{\frac{K_i}{N_i} + \frac{b_{li}}{N_i}s}, \quad (8)$$

where i corresponds to the i th element of the associated diagonal matrix or vector and s denotes the Laplace variable. This solution is obtained by substituting equation (7) into (3) and solving the differential equation for $u(t)$. For the case where we assume the base is perfectly controlled, we solve for the base spatial force by setting $\alpha_B = 0$ and solving for f_b using equation (3). Note that the motor trajectory generation for this case is unchanged from the fixed-base scenario.

For the case where the base is uncontrolled, we must solve for $\alpha_B(t)$ for some given $q_d(t)$ before $u(t)$ can be computed. Using the acceleration propagation detailed in Appendix A, we obtain

$$\alpha_c = \Phi_{cB}\alpha_B + \sum_{k=1}^c \Phi_{ik}(H_k\ddot{q}_k + a_k). \quad (9)$$

We then substitute equation (9) into the top portion of equation (3) and rearrange to solve for the base acceleration, obtaining

$$\alpha_B = (M_B + \Phi_{c,B}^T M_c \Phi_{c,B})^{-1} (-M_{rB}\ddot{q} - C_B - \Phi_{c,B}^T (M_c \sum_{k=1}^c \Phi_{ik}(H_k\ddot{q}_k + a_k) + b_c)). \quad (10)$$

Note that equation (10) is dependent on the base spatial velocity, and that to compute $\alpha_B(t)$ at some given time we must know the base spatial velocity as a function of time, $V_B(t)$. Therefore, the base dynamics must be propagated forward in time from the initial condition to solve for $\alpha_B(t)$. Any suitable integration method may be used to do this - for our work, we use the trapezoidal rule.

Collecting the arm, base, and grasped object dynamics together, we obtain

$$v(t) = M_{Br}\alpha_B + M_r\ddot{q} + C_r + Kq + b_l\dot{q} + J^T f. \quad (11)$$

Note that $f(t)$ is dependent on $\alpha_B(t)$. This signal is once again passed through equation (8) to obtain the correct feedforward motor trajectory for some specified $q_d(t)$.

We next propose the optimization problem to reduce the total trajectory time while satisfying dynamic constraints on joint speed, motor torque, and base spatial force (in the case of a controlled base). As previously stated, we assume the joint-space path ($q_d(\lambda)$) is known and wish to index the path variable with time to generate the trajectory (i.e., find $\lambda(t)$). From equation (3), we find that the link-side trajectory must be sufficiently smooth to obtain finite

values of τ_m when solving the system differential equation. Specifically, $q(t)$ must be differentiable to the third order for systems with coupled linear damping between the motors and joints, and differentiable to the fourth order with systems with decoupled linear damping (i.e., $b_l = 0$). One such trajectory that satisfies the more stringent requirement (and therefore applies to more systems) is the bounded snap (BS) motion profile, which is similar to the commonly used bounded acceleration (trapezoidal velocity) motion profile, but with snap being a constant maximum or minimum value rather than acceleration. We employ the algorithm developed by (Nguyen et al., 2008) to compute this motion profile based on the limits $\eta = [\dot{\lambda}_{\max} \quad \ddot{\lambda}_{\max} \quad \dddot{\lambda}_{\max} \quad \ddot{\lambda}_{\max}]^T$ and obtain $\lambda(t)$. With $\lambda(t)$ and $q(\lambda)$ defined, we can compute $q(t)$.

To compute the motor torque, we must have some estimate of $\ddot{q}_m$. Our choice is to convert equation (6) into a discrete-time system. This does introduce some error, as we treat the motor trajectory signal as discrete time but treat the joint-side trajectory as continuous in simulation to more closely match the physical system. However, we have found that this error is generally small. Other possible choices include estimating $\ddot{q}_m$ via numerical differentiation, which would also introduce numerical error.

Consider the state-space representation of the LPF given by

$$\dot{q}_m = A_L q_m + B_L z, \quad (12)$$

where $z = [q^T \quad \dot{q}^T]^T$. Assuming $q_m(0) = 0$, the motor position at the k th step for the discrete-time system is given by

$$q_{mk} = \sum_{m=0}^{k-1} A_{Ld}^{k-m-1} B_{Ld} z_k, \quad (13)$$

where A_{Ld} , B_{Ld} correspond to the discrete-time versions of A_L , B_L , respectively. From equation (6), we write the state-space representation of the motor dynamics as

$$\dot{x} = Ax + B\tau_m + B_0 z, \quad (14)$$

$$q_m = Cx, \quad (15)$$

where $x = [q_m^T \quad \dot{q}_m^T]^T$ and $C = [\mathbf{I} \quad 0]$, with $\mathbf{I}$ denoting the identity matrix. Once again assuming $q_m(0) = 0$, the discrete time system provides

$$q_{mk} = \sum_{m=0}^{k-1} C A_d^{k-m-1} B_d \tau_m + \sum_{m=0}^{k-1} C A_d^{k-m-1} B_{0d} z, \quad (16)$$

where A_d , B_d , B_{0d} correspond to the discrete-time versions of A , B , and B_0 , respectively.

Let $\bar{\tau}_m$, $\bar{V}$, and $\bar{z}$ be the stacked vectors of each time-step k covering the entire trajectory of $\tau_m(t)$, $v(t)$, and $z(t)$, respectively. Rearranging equations (13) and

Table 1: Assumed Baxter robot parameters

Joint	I_{mi} (kg m ²)	b_{mi} (kg m ² /sec)	N_i
1	0.5E -3.5	8	20
2	0.5E -3.5	8	20
3	0.5E -3.5	8	20
4	0.5E -3.5	8	20
5	0.5E -3.5	6	18
6	0.5E -3.5	6	18
7	0.5E -3.5	6	18

(16) in matrix form for the entire trajectory, we obtain

$$L\bar{V} = H\bar{\tau}_m + P\bar{z}, \quad (17)$$

$$\bar{\tau}_m = H^{-1}(L\bar{V} - P\bar{z}), \quad (18)$$

where L maps the signal $v(t)$ to the motor position through the LPF, H maps the applied motor torque to the motor position, and P maps the joint-side trajectory to the motor position. This provides a way to solve for the motor torque over the entire trajectory.

Finally, we pose the optimization problem

$$\begin{aligned} & \min_{\eta} T_f(\eta) \\ \text{s.t. } & |\tau_m| \preceq \tau_{\max}, \quad |\dot{q}| \preceq \dot{q}_{\max}, \quad |f_b| \preceq f_{\max}, \end{aligned} \quad (19)$$

where $T_f(\eta)$ denotes the final time of the trajectory. We note that $T_f(\eta)$ is obtained via a numerical algorithm in (Nguyen et al., 2008), so we omit the computation of $T_f(\eta)$ here for brevity and direct the reader to the citation. We propose solving this optimization problem using an iterative constrained non-linear optimization method, such as sequential quadratic programming (SQP).

5 Experimental Setup

5.1 Experimental Setup: Computer Simulation

We choose the Rethink Robotics Baxter robot (Rethink Robotics, 2018) as a stand-in for a space manipulator in our simulation. This robot has 7 DOF and series-elastic actuators, meaning that it exhibits significant joint-flexibility. The link mass and inertia, link COM, and joint spring coefficients are all available from (Rethink Robotics, 2018). We choose the computer simulation motor inertias, motor damping, joint damping, and gear ratios as shown in Table 1.

We choose the satellite parameters available at (Kramer, 2002), and model both the target and base satellites as cylinders with evenly distributed mass,

Table 2: Assumed base and target satellite parameters

Satellite	mass (kg)	radius (m)	length (m)
Target	2,200	0.7	4.3
Base	6,600	1.5	5

with parameters provided in Table 2. Base parameters are chosen at a similar scale as the target satellite, but slightly larger, with parameters provided in Table 2. Both the robot base and grasp point are located on the flat surfaces of the cylindrical satellites, as shown in Figure 3. There is a 1 m offset from the center axis of the base satellite to the first joint of the robot arm and a 0.5 m offset from the center axis of the target satellite to the grasp point.

Full spatial simulations are conducted with MATLAB and the Simscape Multibody package. A separate dynamic simulation that employs the Newton-Euler algorithm to solve the equations of motion is also implemented using MATLAB and Simulink. The results of these simulations have been compared to confirm the Multibody simulation results match the dynamics shown in equation (3). The optimization problem is solved using the SQP algorithm with the built-in MATLAB *fmincon* function.

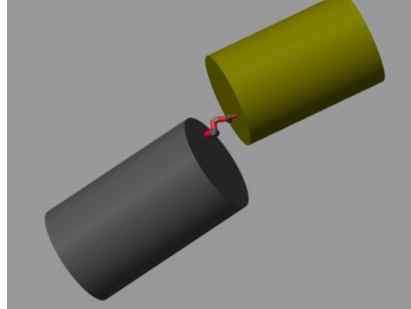


Figure 3: Simulation model of Baxter arm with target satellite (gold) and base satellite (gray).

5.2 Experimental Setup: Physical Experiment

To investigate the efficacy of the feedforward motor trajectory for oscillation reduction, we construct a single flexible joint for our physical experiment. The joint is composed of a TowerPro MG995R servo motor, a 25.4 mm length section of ~ 9 mm diameter 60A durometer polyurethane, and a rigid appendage connected to a second air bearing object, as shown in Figure 4. We mount this joint to two air bearing objects, which use three 40 mm diameter Newway air

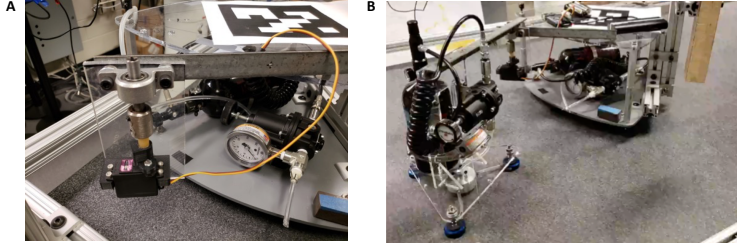


Figure 4: Flexible joint experimental setup: **(A)** detailed view of custom flexible joint, **(B)** air bearing bodies connected via flexible joint.

bearings each to reduce friction between the objects and a Mojave granite table. On-board air is supplied using compressed gas cylinders, as shown in Figure 4. The air supply to the base object is turned off during our experiments, making the system a fixed-base 1 DOF robot. The floating air bearing object has a mass of approximately 4.1 kg.

We use two fiducial tags, one mounted on each air bearing object, to estimate the link-side joint position. Computer vision data is taken using a 1.3 mega-pixel FLIR color Flea3 machine vision camera at approximately 20 Hz. The servo motor is controlled via a Raspberry Pi 3 with a servo control hat. We assume that the servo motor control loop is tight, and therefore that the commanded servo motor position is equivalent to the actual servo motor position. The Raspberry Pi code and computer vision code are integrated using Robot Raconteur (RR) (Wason, 2016) at approximately 20 Hz so that time-stamped vision and motor data can be recorded. We slow the command/sample rate using the same strategy as (Peng et al., 2018) to achieve a consistent 20 Hz.

6 Experimental Results

6.1 Simulation Experiment Results

First, we check the efficacy of the feedforward motor trajectory given by equation (8) to remove oscillations in simulation. As our model knowledge is complete, all oscillations should be successfully removed. However, modeling error and disturbances will require additional feedback control in a real physical system. We investigate the feedforward motor trajectory in part to ensure our derivations and algorithms for generating $u(t)$ are correct. Furthermore, we compare the feedforward motor trajectory to the case where rigid joints are assumed (denoted the naïve case; $u(t) = Nq_d(t)$). Note that the naïve method suffices when joints are sufficiently rigid, and tracking is perfect in the infinitely stiff (rigid) joint case. This comparison shows the significance of the flexible joint dynamics in the space robot application, and what issues may arise if rigid joints are assumed during trajectory generation or trajectory following.

We first compare simulation results for a fixed-base scenario. A desired link-side BS trajectory is generated for this test case using $\eta = [0.04 \ 0.05 \ 0.02 \ 0.3]^T$, with a final time of approximately 27.9 s. We use a previously generated path, $q_d(\lambda)$, for all cases considered, which involves 6-DOF motion of the target object and motion of all of the 7 robot joints.

The joint-space performance results are provided in Figure 5. It is clear from these results that the naïve motor trajectory is not sufficient, with oscillations continuing well after the trajectory has been completed (at ~ 30 sec), even in cases where position tracking had decent accuracy during desired motion, such as joints 2, 3, 4, and 6. These oscillations are particularly dangerous, as they correspond to link-side motion after the motors have reached steady-state. Physically, this is analogous to a “brakes-on” condition, meaning that simply applying motor brakes at the end of the naïve motor trajectory will lead to significant oscillations and motion that may damage the robot arm or lead to collisions between the satellites. In comparison, the motor trajectory generated using the LPF approach has good joint-space performance. We do note that choosing a large zero-order hold (ZOH) step size for the motor trajectory can lead to oscillations even when using the LPF approach, especially for faster trajectories. This should be a consideration for physical implementation of this algorithm. Comparison results are presented for a 400 Hz motor control loop.

Next, we consider the importance of accounting for the floating base dynamics. Once again, we investigate this scenario in part to confirm our derivations and algorithms for generating $u(t)$. In cases where the robot base satellite is far more massive than the target satellite, and the robot arm itself, the base dynamics can be ignored and the robot can be treated as a fixed-base manipulator. However, in cases where the base is not sufficiently massive the base dynamics can have a large impact on the motion of the manipulator arm and captured satellite, as we will show. We compare motor trajectory generation without accounting for the base dynamics (using equations (7) and (8)) against trajectory generation accounting for the base dynamics (using equations (11) and (8)) in Figure 6. The BS trajectory limits used in the fixed-base example are used again here. These results show that failure to account for the uncontrolled base dynamics can result in poor tracking performance and oscillations, even when using the LPF trajectory generation approach. Accounting for the base dynamics leads to good tracking performance, showing the importance of taking base dynamics into account when joint flexibility and a payload of comparable mass to the base are both present.

With a methodology for generating $u(t) = q_m(t)$ given some $q_d(t)$, the next question is how to generate $q_d(t)$ itself subject to the constraints previously outlined. We utilize the *fmincon* function within MATLAB to implement equation (19) with an SQP algorithm. An uncontrolled floating base is assumed for these trajectory generation scenarios. We slow the assumed motor control loop to 50 Hz to reduce the computational requirement for checking dynamic constraints, with small loss of accuracy in the computation of motor torque and base actuation. Furthermore, we choose $\tau_{\max} = [75 \ 75 \ 75 \ 75 \ 50 \ 50 \ 50]^T$ Nm.

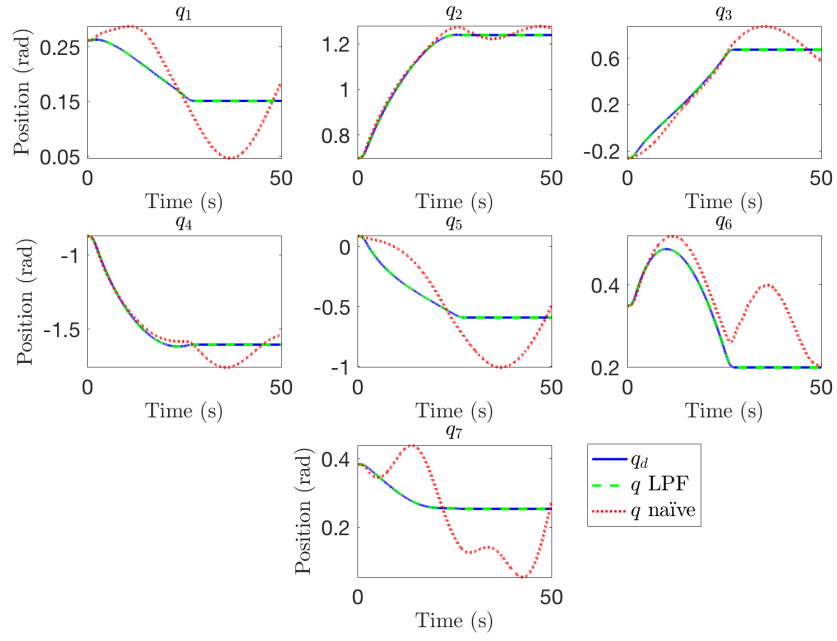


Figure 5: Comparison of joint space performance for a naïve motor trajectory and a trajectory generated using the LPF method.

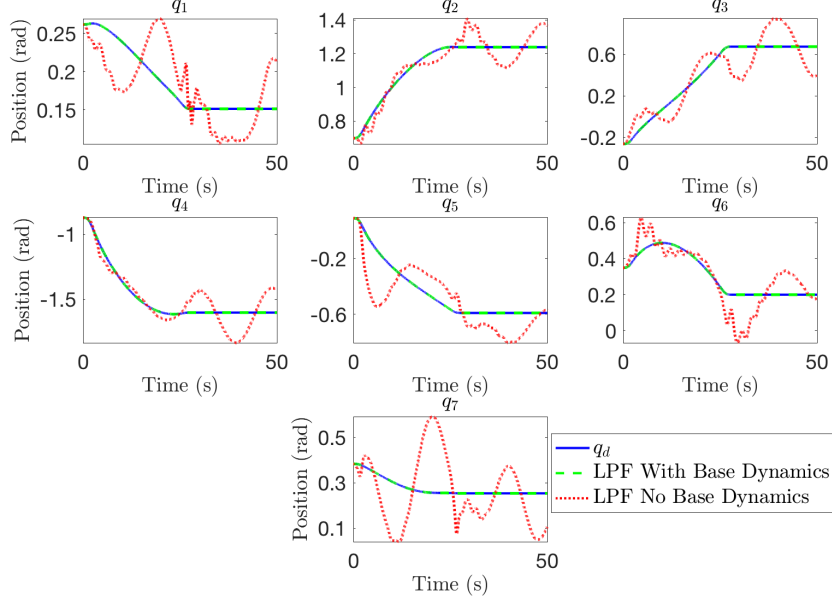


Figure 6: Comparison of joint space performance with uncontrolled floating base when base dynamics are and are not accounted for.

The BS trajectory limits used in the previous example are treated as the initial guess, η_0 . Figure 7 presents motor torque results for an optimized trajectory obtained for an uncontrolled floating base scenario. As these results show, joint 5 reaches the torque limit. A final trajectory time of 12.2 s was obtained, which is a 56% improvement over the initial guess, and illustrates the benefits of undertaking the optimization step.

We also examine a case with link-side joint speed limits, and add the constraint $\dot{q}_{\max} = 0.1$ rad/s for each joint to the previously considered case. Figure 8 shows the joint speed limits for this case. We see that joint 4 reaches the joint speed limit, showing how different joints may be the limiting factor for the same path depending on the dynamic constraints. A final trajectory time of 19.7 s, a 29% improvement over the initial guess, was obtained. These results show the efficacy of the trajectory generation for reducing total trajectory time subject to dynamic constraints.

Finally, we consider the case with a controlled floating base, and choose $f_{\max} = (1500, 200)$ Nm, N for base actuation torque and force limits, respectively. Figure 9 shows the base actuation required to keep the base stationary during the generated trajectory, where the y direction force hits the specified limit. The final trajectory time is 16.0 s, which is a 43% improvement.

It is important to note that feedback control will be needed on each joint

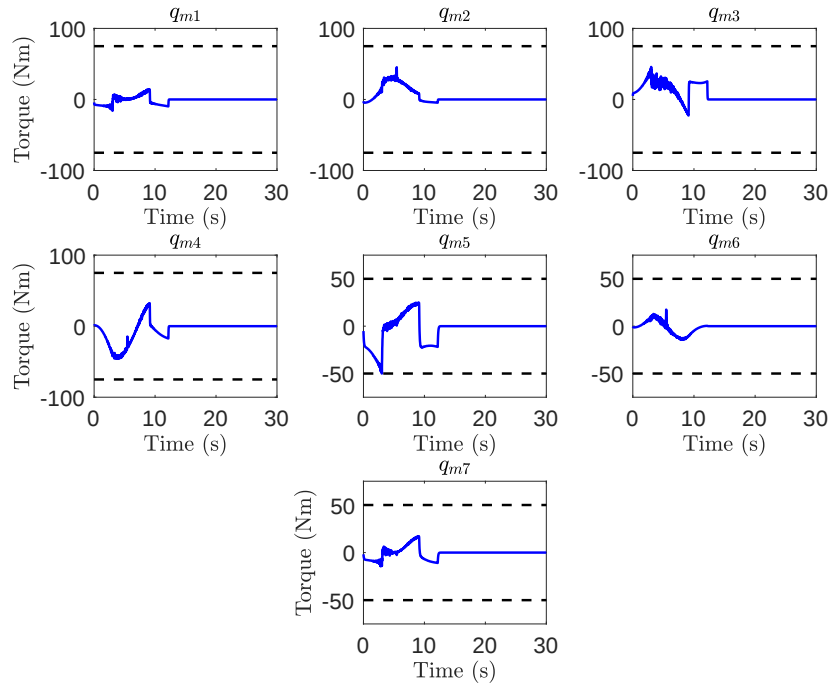


Figure 7: Simulated motor torques for the uncontrolled floating base trajectory after optimization. Note that joint 5 reaches the lower torque limit.

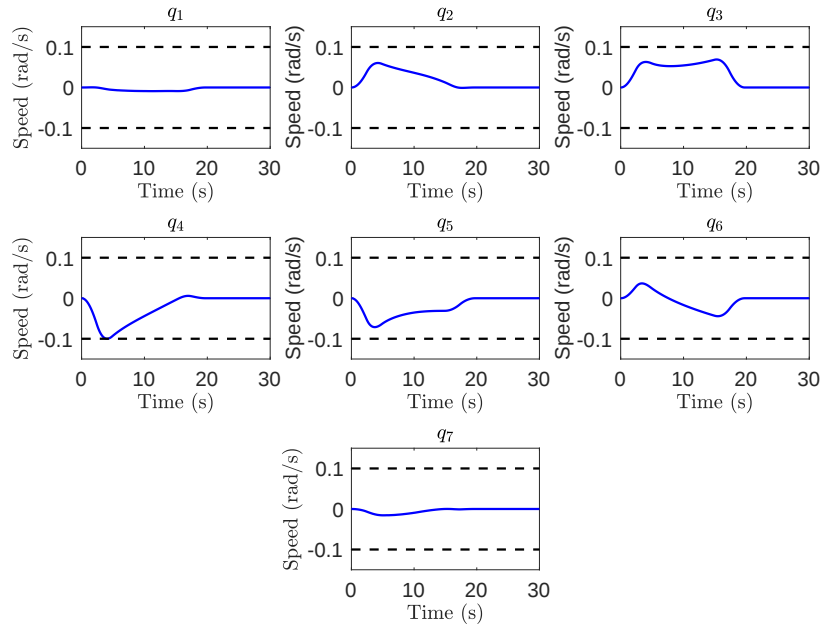


Figure 8: Joint speeds for an uncontrolled floating base scenario after optimization where joint speed is the limiting factor. Note that joint 4 reaches the lower joint speed limit.

and on the base actuators (if controlled) on-application. Therefore, the actual joint torque and base actuation will likely be higher. It is therefore important to include a safety margin when specifying dynamic limits. One method is simply to set the safety margin as a percentage of the total limit, e.g., setting a 50% margin on a motor saturation of 5 Nm would generate a torque limit for trajectory generation of 2.5 Nm.

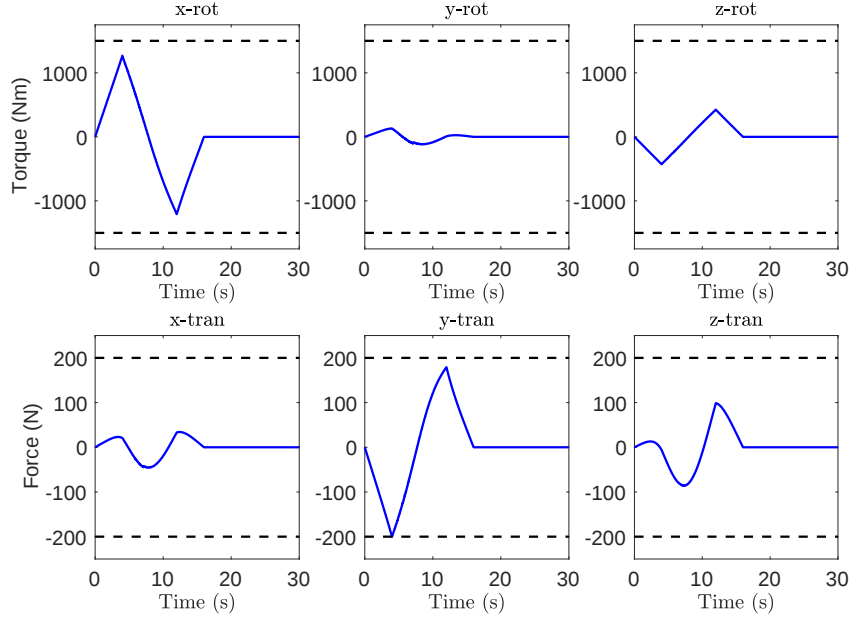


Figure 9: Feedforward base actuation during controlled base trajectory after optimization. Note that the y -direction force reaches the specified limit.

6.2 Physical Experiment Results

We include the physical experiments to show the efficacy of providing a feedforward motor trajectory to reduce oscillations during trajectory following. This feedforward trajectory can be used in tandem with one of the flexible joint control options described in the Introduction to further reduce oscillations. We first perform model identification on our physical testbed by finding the frequency response to a known Schroeder-phase input signal (Bayard, 1993). Based on the physical model of a single flexible joint (Figure 2), we fit the frequency response to a linear transfer function with the following two constraints: zeros must be in the left half plane, and the DC gain is 1. We obtain the linear transfer function

$G(s)$ from $q_m(s)$ to $q(s)$ given by

$$G(s) = \frac{q(s)}{q_m(s)} = \frac{0.0695}{0.281s^2 + 0.0322s + 0.0695}. \quad (20)$$

We note that this transfer function fits a model with negligible coupled damping between the motor and link sides (i.e., $b_l = 0$). This corresponds with little damping in the flexible transmission of the physical system. Figure 10 shows the frequency response of $G(s)$ compared against the experimental data, showing the accuracy of the model fit.

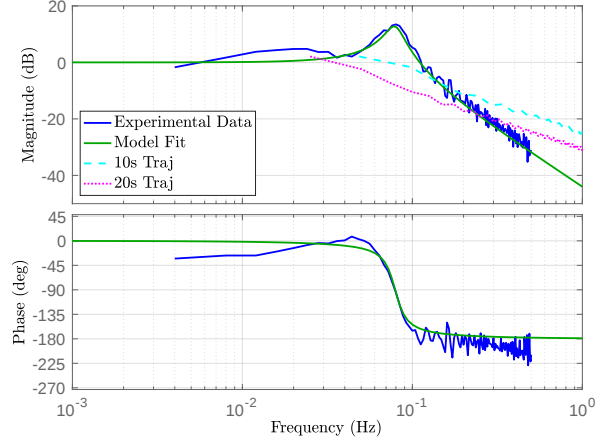


Figure 10: Frequency response of model fit, experimental data, and motor trajectories.

We employ a similar methodology as detailed above to generate the feed-forward motor trajectory, and invert $G(s)$ to generate $q_m(t)$ given $q_d(t)$ (i.e., solving the differential equation for $q_m(t)$). Note that inversion is possible even though $G(s)$ is a proper transfer function because $q_d(t)$ is known $\forall t$.

We investigate two test cases and repeat all experiments 5 times: a 10 sec BS and 20 sec BS motion profile for $q_d(t)$. Figure 11 provides representative results for both of these cases, while Table 3 provides the mean root-mean-squared-error (RMSE) and standard deviation (STD) of the RMSE over all trials. From these results, we see a clear reduction in oscillations and better tracking performance for the 10 sec trajectory case. Furthermore, all 5 trials using the naïve choice ($q_m(t) = q_d(t)$) for the 10 sec trajectory resulted in collision between the two air bearing objects. No collisions occurred when using the model inversion approach.

However, we do not see significant improvement for the 20 sec trajectory case. Figure 10 shows the magnitude response of the desired link-side trajectory for both trajectory cases. This result suggests that the 20 sec trajectory case does

Table 3: Mean RMSE and STD of trajectory error

Trajectory	10s Naïve	10s Model Inv.	20s Naïve	20s Model Inv.
RMSE (rad)	0.308	0.242	0.107	0.082
STD	0.03	0.02	0.01	0.003

not excite the flexible mode at ~ 0.075 Hz, whereas the 10 sec trajectory case does. Therefore, there is little benefit to using the feedforward motor trajectory for oscillation reduction in the 20 sec case. This also suggests that slowing the trajectory may be sufficient to avoid oscillations with joint flexibility and the naïve feedforward approach, at the cost of total trajectory time. Finally, the feedforward motor trajectory is not enough to completely remove oscillations, and the tracking performance can still be improved. This is to be expected due to modeling error, unmodeled dynamics, and environmental disturbances. Future work will investigate using feedback control on top of the presented feedforward methodology to further reduce oscillations and improve tracking performance.

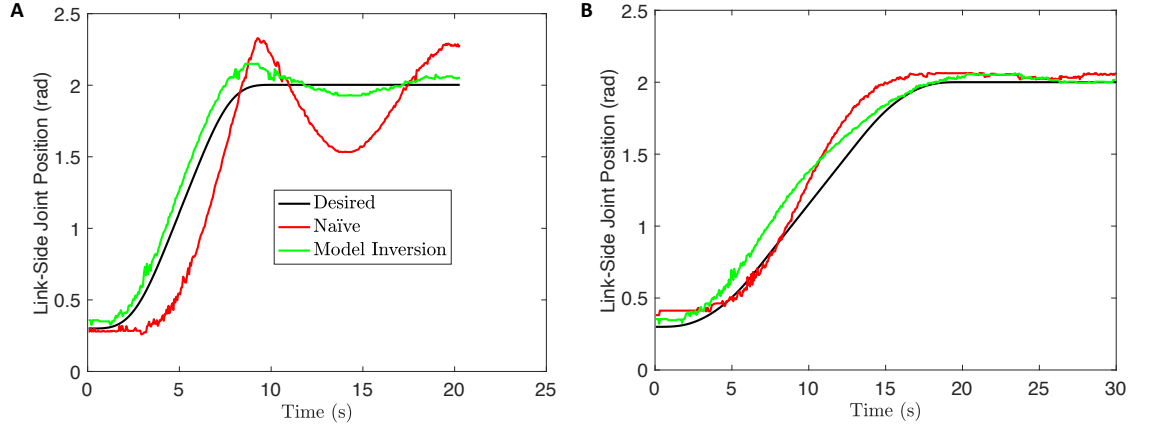


Figure 11: Single joint experimental results: (A) 10 sec trajectory, (B) 20 sec trajectory.

7 Conclusions and Future Work

This paper provides a model-based methodology for trajectory generation for a flexible-joint space manipulator grasping a massive object. We assume that the joint-space path has already been generated, and that trajectory generation indexes the path variable with time subject to dynamic constraints on motor torque, joint speed, and floating base actuation. Furthermore, we consider cases with both an uncontrolled floating base and a controlled floating base. Methodologies are provided to generate a feedforward motor trajectory given a desired link-side trajectory, and to generate a desired link-side trajectory while minimizing total trajectory time subject to dynamic constraints.

Full-spatial simulation results show the efficacy of this trajectory generation method. We also present physical experiment results for a single-joint scenario, showing that the feedforward motor trajectory can be used to improve tracking performance and oscillation reduction for certain trajectories in physical systems as well. However, some oscillations and tracking error remain, as expected. The addition of existing feedback controllers for flexible joint systems would likely further reduce these oscillations. The results also suggest that flexible modes may be avoided by using slower trajectories, albeit at the cost of trajectory time.

Future work will incorporate feedback control for further vibration suppression and tracking performance improvement. Motor control dynamics can also be considered for higher fidelity trajectory generation. Furthermore, future work will investigate more complex scenarios on the air bearing table. These scenarios will include controlled and uncontrolled floating bases, as well as higher DOF robots. We also note the assumption of a pre-planned path may impose limitations on trajectory optimization. In particular, redundancy resolution cannot be leveraged during optimization, and certain paths may make satisfying dynamic constraints more difficult (e.g., large path curvature at a weaker wrist joint). These concerns should be addressed by allowing the joint-space path to be iteratively modified. Future work will investigate these improvements to the trajectory generation methodology, as well as incorporate practical kinematic path generation concerns like obstacle avoidance, joint limit constraints, and singularity avoidance.

The trajectory parameterization choice may also limit optimization. We have chosen a small number of optimization variables to parameterize the trajectory (a total of four), but this may result in certain portions of the trajectory being artificially slowed. A small portion of the path (e.g., one with high curvature with respect to the path variable) may also limit the optimized trajectory by reaching dynamic constraints at lower velocities and accelerations than other portions of the path would reach constraints at. Future work should investigate other trajectory parameterizations, such as polynomial or piece-wise definitions, which can allow for more complex trajectory shapes and for portions of the path to be optimized separately. We do note that computation time is a practical concern, so any parameterization choice with additional complexity should be evaluated for speed of computation as well. Piece-wise definitions of the trajec-

tory should also take care to not introduce discontinuities or high acceleration, jerk, or snap in transition regions, as these could lead to oscillatory behavior or difficulties in satisfying dynamic constraints.

Conflict of Interest Statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author Contributions

DC and JW: study concept and design, development of methodology, analysis of results, and manuscript preparation.

DC: simulation and physical experiment setup, data collection.

Funding

The authors gratefully acknowledge the support of this project by the United States government. The project was also supported in part by the New York State Empire State Development Division of Science, Technology and Innovation (NYSTAR) under NYSTAR contract C130145.

Acknowledgments

The authors would like to thank Glenn Saunders and Kenneth Myer for their help in designing and fabricating the experiment test fixtures, as well as Craig Carignan and Billy Gallagher for helpful discussions on robotic satellite servicing.

References

- Alberts, T. E., Xia, H., and Chen, Y. (1992). Dynamic analysis to evaluate viscoelastic passive damping augmentation for the Space Shuttle Remote Manipulator System. *ASME J. Dyn. Syst. Meas. and Control* 114, 468–475. doi:<https://doi.org/10.1115/1.2897370>
- Bayard, D. (1993). Statistical plant set estimation using Schroeder-phased multisinusoidal input design. *Appl. Math. and Comput.* 58, 169–198. doi:[https://doi.org/10.1016/0096-3003\(93\)90135-2](https://doi.org/10.1016/0096-3003(93)90135-2)
- Bell, E. (2020). Nasa space science data coordinated archive. NASA. <https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=2007-006A>;

- <https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=2007-006C> [Accessed Mar. 5, 2020]
- Bobrow, J. E., Dubowsky, S., and Gibson, J. (1985). Time-optimal control of robotic manipulators along specified paths. *Int. J. Robot. Res.* 4, 3–17. doi:<https://doi.org/10.1177/027836498500400301>
- Carabis, D. (2020). *Robotic manipulation of massive objects in space*. Ph.D. thesis, Dept. Mech. Aerosp. Nucl. Eng., Rensselaer Polytechnic Institute, Troy, NY, USA
- Carabis, D. S., Oakes, K. I., and Wen, J. T. (2021). Manipulation of massive objects in space using flexible joint manipulators. *J. Guid. Control and Dyn.*, 1–15doi:<https://doi.org/10.2514/1.G005347>
- Chang, K. (2020). An orbital rendezvous demonstrates a space junk solution. The New York Times. <https://www.nytimes.com/2020/02/26/science/mev-1-northrop-grumman-space-junk.html> [Accessed Mar. 31, 2020]
- De Luca, A., Lucibello, P., and Ulivi, G. (1989). Inversion techniques for trajectory control of flexible robot arms. *J. Robot. Syst.* 6, 325–344. doi:<https://doi.org/10.1002/rob.4620060403>
- Dubowsky, S., Vance, E. E., and Torres, M. A. (1989). The control of space manipulators subject to spacecraft attitude control saturation limits. In *Proc. NASA Conf. on Space Telerobotics*. 409–418
- Ferretti, G., Magnani, G., Rocco, P., Viganò, L., and Rusconi, A. (2005). On the use of torque sensors in a space robotics application. In *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* 1947–1952. doi:<https://doi.org/10.1109/IROS.2005.1545374>
- Flores-Abad, A., Ma, O., Pham, K., and Ulrich, S. (2014). A review of space robotics technologies for on-orbit servicing. *Prog. Aerosp. Sci.*, 1–26doi:<https://doi.org/10.1016/j.paerosci.2014.03.002>
- Garner, R. (2018). About the Hubble space telescope. NASA. https://www.nasa.gov/mission/_pages/hubble/story/index.html [Accessed Dec. 1, 2018]
- Hall, L. (2019). Northrop grumman and nasa partnership gives satellites a new lease on life. NASA. https://www.nasa.gov/directorates/heo/NASA_Partnership_Gives_Satellites_a_New_Lease_on_Life [Accessed Mar. 31, 2020]
- Huang, P. and Xu, Y. (2006). PSO-based time-optimal trajectory planning for space robot with dynamic constraints. In *IEEE Int. Conf. on Robot. and Biomimetics*. 1402–1407. doi:<https://doi.org/10.1109/ROBIO.2006.340134>

- Jain, A. (2010). *Robot and Multibody Dynamics: Analysis and Algorithms* (New York, NY, USA: Springer Science and Business Media)
- Kasai, T., Oda, M., and Suzuki, T. (1999). Results of the ETS-7 mission- rendezvous docking and space robotics experiments. In *Proc. Int. Symp. Artif. Intell. Robot. and Automat. in Space*. vol. 440, 299–306
- Kauderer, A. (2012). Canadarm2 to Catch SpaceX’s Dragon on Its Maiden Voyage to the ISS. NASA.gov. https://www.nasa.gov/mission/_pages/station/main/dragon/_catch.html [Accessed Mar. 5, 2020]
- Kramer, H. J. (2002). Landsat-7. European Space Agency. <https://earth.esa.int/web/eoportal/satellite-missions/1/landsat-7> [Accessed Jul. 24, 2019]
- Ma, O., Buhariwala, K., Roger, N., MacLean, J., and Carr, R. (1997). MDSF — a generic development and simulation facility for flexible, complex robotic systems. *Robotica* 15, 49–62. doi:<https://doi.org/10.1017/S0263574797000076>
- Moosavian, S. A. A. and Papadopoulos, E. (2007). Free-flying robots in space: An overview of dynamics modeling, planning and control. *Robotica* 25, 537–547. doi:<https://doi.org/10.1017/S0263574707003438>
- Nanos, K. and Papadopoulos, E. G. (2015). On the dynamics and control of flexible joint space manipulators. *Control Eng. Pract.* 45, 230–243. doi: <https://doi.org/10.1016/j.conengprac.2015.06.009>
- Nguyen, K. D., Ng, T.-C., and Chen, I.-M. (2008). On algorithms for planning s-curve motion profiles. *Int. J. Adv. Robot. Syst.* 5, 99–106. doi:<https://doi.org/10.5772/5652>
- Northrop Grumman Space Logistics (2020). Mev-1 mission profile. Northrop Grumman. <https://www.northropgrumman.com/space/space-logistics-services/mission-extension-vehicle.html> [Accessed Mar. 31, 2020]
- Ogilvie, A., Allport, J., Hannah, M., and Lymer, J. (2008). Autonomous satellite servicing using the Orbital Express Demonstration Manipulator System. In *9th Int. Symp. on Artif. Intell. Robot. and Automat. in Space*. 25–29
- Peng, Y.-C., Carabis, D. S., and Wen, J. T. (2018). Collaborative manipulation with multiple dual-arm robots under human guidance. *Int. J. Intell. Robot. and Appl.* 2, 252–266. doi:<https://doi.org/10.1007/s41315-018-0053-y>
- Rethink Robotics (2018). Rethink Robotics Baxter Arms SDK Wiki. RethinkRobotics.com. <http://sdk.rethinkrobotics.com/wiki/arms> [Accessed Mar. 7, 2018]
- Rybus, T. and Seweryn, K. (2016). Planar air-bearing microgravity simulators: Review of applications, existing solutions and design parameters. *Acta Astronautica* 120, 239–259. doi:<https://doi.org/10.1016/j.actaastro.2015.12.018>

- Scott, M. A., Gilbert, M. G., and Demeo, M. E. (1993). Active vibration damping of the Space Shuttle Remote Manipulator System. *J. Guid. Control and Dyn.* 16, 275–280. doi:<https://doi.org/10.2514/3.56598>
- Spong, M. W. (1987). Modeling and control of elastic joint robots. *J. Dyn. Syst. Meas. and Control* 109, 310–318. doi:<https://doi.org/10.1115/1.3143860>
- Spong, M. W. (1989). Adaptive control of flexible joint manipulators. *Syst. and Control Lett.* 13, 15–21. doi:<https://doi.org/10.1109/37.41450>
- Sun, L., Yin, W., Wang, M., and Liu, J. (2018). Position control for flexible joint robot based on online gravity compensation with vibration suppression. *IEEE Trans. Ind. Electron.* 65, 4840–4848. doi:<https://doi.org/10.1109/TIE.2017.2772157>
- Tasker, F. and Henshaw, C. G. (2008). Managing contact dynamics for orbital robotic servicing missions. In *AIAA Space Conf.* (San Diego, CA, USA), 7908–7924. doi:<https://doi.org/10.2514/6.2008-7908>
- Ulrich, S. and Sasiadek, J. Z. (2012a). Real-time estimation and adaptive control of flexible joint space manipulators. In *Robot Motion and Control* (Springer). 215–224. doi:https://doi.org/10.1007/978-1-4471-2343-9_18
- Ulrich, S., Sasiadek, J. Z., and Barkana, I. (2012). Modeling and direct adaptive control of a flexible-joint manipulator. *J. Guid. Control and Dyn.* 35, 25–39. doi:<https://doi.org/10.2514/1.54083>
- Ulrich, S. and Sasiadek, Z. (2012b). Trajectory tracking control of flexible-joint space manipulators. *Can. Aeronaut. and Space J.* 58, 47–59. doi:<https://doi.org/10.5589/q12-004>
- Wason, J. D. (2016). Robot Raconteur® version 0.8: An updated communication system for robotics, automation, building control, and the internet of things. In *IEEE Int. Conf. on Automat. Sci. and Eng.* 595–602. doi:<https://doi.org/10.1109/COASE.2016.7743456>
- Wu, H., Sun, F., Sun, Z., and Wu, L. (2004). Optimal trajectory planning of a flexible dual-arm space robot with vibration reduction. *J. Intell. and Robot. Syst.* 40, 147–163. doi:<https://doi.org/10.1023/B:JINT.0000038946.21921.c7>
- Yamano, M., Konno, A., and Uchiyama, M. (2000). Experiments on capturing a floating object by two flexible manipulators. In *IEEE International Conference on Robotics and Automation, 2000* (IEEE), vol. 1, 482–487. doi:<https://doi.org/10.1109/ROBOT.2000.844101>
- Yin, W., Sun, L., Wang, M., and Liu, J. (2018). Nonlinear state feedback position control for flexible joint robot with energy shaping. *Robot. and Auton. Syst.* 99, 121–134. doi:<https://doi.org/10.1016/j.robot.2017.10.007>

- Yoshida, K. (2003). Engineering Test Satellite VII flight experiments for space robot dynamics and control: Theories on laboratory test beds ten years ago, now in orbit. *Int. J. Robot. Res.* 22, 321–335. doi:<https://doi.org/10.1177/0278364903022005003>
- Yu, X. and Chen, L. (2014). Dynamic modeling and control of a free-flying space robot with flexible-link and flexible-joints. In *IEEE Int. Conf. on Robot. and Automat.* 6625–6630. doi:<https://doi.org/10.1109/ICRA.2014.6907837>
- Zarafshan, P. and Moosavian, S. A. A. (2011). Control of a space robot with flexible members. In *IEEE Int. Conf. on Robot. and Automat.* 2211–2216. doi:<https://doi.org/10.1109/ICRA.2011.5979882>

Appendix A: Model Derivation

This Appendix details the derivation of the flexible-joint space manipulator with floating base model. The flexible joint model is based on (Spong, 1987), while the spatial algebra approach (utilizing the Newton-Euler method) used to derive the arm dynamics is based on (Jain, 2010).

We make two key assumptions in our derivation: that the motors can be modeled by 1-D rotational inertias, and that the kinetic energy of the motors is due only to their own rotation relative to the stator (i.e., we ignore the inertial coupling between the motors and the arm links). The motor dynamics are then easily derived through applying Euler’s laws, and are presented in Section 3. Similarly, the joint torque τ due to the flexible connection between the motor and link is easily derived and presented in Section 3.

We will consider a serial chain with only rotation joints, with $i = B, 1, \dots, n$. Let h_i, f_i be the axis of rotation of the i th joint and the spatial force at the i th joint, represented in the i th joint frame. Define

$$H_i \in \mathbb{R}^6 = \begin{bmatrix} h_i \\ 0 \end{bmatrix}, \quad \forall i = 1, \dots, n. \quad (\text{A.1})$$

Therefore,

$$\tau_i = H_i^T f_i. \quad (\text{A.2})$$

Note that $H_B = \mathbf{I} \in \mathbb{R}^6$, where $\mathbf{I}$ is the identity matrix.

We first consider the boundary conditions. Assuming negligible gravitational effects, we set the boundary conditions as

$$V_0 = 0, \quad \alpha_0 = 0, \quad f_{n+1} = f, \quad (\text{A.3})$$

where V_0 and α_0 are the spatial velocity and acceleration boundary conditions, respectively. Here, $n + 1$ denotes the end effector frame.

Next, we consider the spatial velocity and acceleration propagation. The spatial velocity propagation is given by

$$V_{i+1} = \Phi_{i+1,i} V_i + H_{i+1} \dot{q}_{i+1}, \quad (\text{A.4})$$

where

$$\Phi_{i+1,i} = \begin{bmatrix} R_{i+1,i} & 0 \\ -R_{i+1,i} p_{i,i+1}^\times & R_{i+1,i} \end{bmatrix}. \quad (\text{A.5})$$

Here, R_{ij} denotes the rotation matrix from the i th frame to the j th frame, and p_{ij} denotes the vector from the i th frame to the j th frame, represented in the i th frame. Note that $\Phi_{i+1,i}$ is therefore dependent on q . The spatial acceleration propagation is given by

$$\alpha_{i+1} = \Phi_{i+1,i} \alpha_i + H_{i+1} \ddot{q}_{i+1} + a_{i+1}, \quad (\text{A.6})$$

where

$$a_{i+1} = \begin{bmatrix} (R_{i+1,i} \omega_i)^\times \omega_{i+1} \\ R_{i+1,i} \omega_i^\times \omega_i^\times p_{i,i+1} \end{bmatrix}, \quad (\text{A.7})$$

for robots with all rotational joints. The rotational velocity of the i th frame is given by ω_i . Note that $a_B = 0$.

The Newton-Euler equations for the i th body provide

$$M_i \alpha_i + b_i = f_i - \Phi_{i+1,i}^T f_{i+1}, \quad (\text{A.8})$$

where

$$M_i = \begin{bmatrix} (I_i)_i & m_i p_{i,c}^\times \\ -m_i p_{i,c}^\times & m_i \mathbf{I} \end{bmatrix}, \quad (\text{A.9})$$

and

$$b_i = \begin{bmatrix} \omega_i^\times (I_i)_i \omega_i \\ m_i \omega_i^\times \omega_i^\times p_{i,c} \end{bmatrix}. \quad (\text{A.10})$$

The inertia matrix of the i th link at and represented in the i th coordinate frame is given by $(I_i)_i$, while the mass of the i th link is denoted by m_i . The vector from the i th frame to the i th link COM, represented in the i th frame, is given by $p_{i,c}$.

We can then generalize these equations, noting that the spatial acceleration boundary condition is 0, and obtain

$$\alpha_i = \sum_{k=B}^i \Phi_{ik} (H_k \ddot{q}_k + a_k), \quad (\text{A.11})$$

and

$$f_i = \Phi_{n+1,i}^T f + \sum_{k=i}^n \Phi_{ki}^T (M_k \alpha_k + b_k). \quad (\text{A.12})$$

Define the stacked vectors and block matrices

$$\alpha = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}, \quad \bar{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}, \quad a = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix},$$

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \Phi_{n+1,n}^T \end{bmatrix},$$

$$H = \begin{bmatrix} H_1 & & 0 \\ & \ddots & \\ 0 & & H_n \end{bmatrix}, \quad M = \begin{bmatrix} M_1 & & 0 \\ & \ddots & \\ 0 & & M_n \end{bmatrix},$$

and

$$\Phi = \begin{bmatrix} \Phi_{11} & & 0 \\ \vdots & \ddots & \\ \Phi_{n1} & \dots & \Phi_{nn} \end{bmatrix}.$$

These vectors and matrices are used to derive the standard manipulator arm (fixed base) dynamics. We include them to show the relationship between the floating base and fixed base models.

Including the base, define the block matrices

$$\alpha^* = \begin{bmatrix} \alpha_B \\ \alpha \end{bmatrix}, \quad f^* = \begin{bmatrix} f_B \\ \bar{f} \end{bmatrix}, \quad a^* = \begin{bmatrix} 0 \\ a \end{bmatrix},$$

$$b^* = \begin{bmatrix} b_B \\ b \end{bmatrix}, \quad B^* = \begin{bmatrix} 0 \\ B \end{bmatrix},$$

$$H^* = \begin{bmatrix} H_B & 0 \\ 0 & H \end{bmatrix}, \quad M^* = \begin{bmatrix} M_B & 0 \\ 0 & M \end{bmatrix},$$

and

$$\Phi^* = \begin{bmatrix} \Phi_{BB} & \dots & 0 \\ \vdots & & \Phi \\ \Phi_{nB} & & \end{bmatrix}.$$

Therefore,

$$\alpha^* = \Phi^*(H^* \begin{bmatrix} \alpha_B \\ \ddot{q} \end{bmatrix} + a^*), \quad (\text{A.13})$$

and

$$f^* = \Phi^{*T} B^* f + \Phi^{*T} (M^* \alpha^* + b^*) \quad (\text{A.14})$$

Substituting equation (A.13) into equation (A.14) yields

$$H^{*T} f^* = \begin{bmatrix} f_B \\ \tau \end{bmatrix} = H^{*T} \Phi^{*T} B^* f + H^{*T} \Phi^{*T} (M^* \Phi^* (H^* [\alpha_B \ddot{q}] + a^*) + b^*). \quad (\text{A.15})$$

Simplification and combining block matrices conveniently yields

$$\begin{bmatrix} f_B \\ \tau \end{bmatrix} = \begin{bmatrix} \Phi_{nB}^T \Phi_{n+1,n}^T \\ H^T \Phi^T B \end{bmatrix} f + H^{*T} \Phi^{*T} (M^* \Phi^* (\begin{bmatrix} \alpha_B \\ H \ddot{q} + a \end{bmatrix}) + b^*),$$

$$\begin{aligned}\begin{bmatrix} f_B \\ \tau \end{bmatrix} &= \begin{bmatrix} \Phi_{n+1,B}^T \\ H^T \Phi^T B \end{bmatrix} f + H^{*T} \Phi^{*T} (M^* \begin{bmatrix} \alpha_B \\ \hat{\Phi}_1 \alpha_B + \Phi H \ddot{q} + \Phi a \end{bmatrix} + b^*), \\ \begin{bmatrix} f_B \\ \tau \end{bmatrix} &= \begin{bmatrix} \Phi_{n+1,B}^T \\ H^T \Phi^T B \end{bmatrix} f + H^{*T} \Phi^{*T} \begin{bmatrix} M_B \alpha_B \\ M \hat{\Phi}_1 \alpha_B + M \Phi H \ddot{q} + M \Phi a \end{bmatrix} + b^*),\end{aligned}$$

and

$$\begin{bmatrix} f_B \\ \tau \end{bmatrix} = \begin{bmatrix} \Phi_{n+1,B}^T \\ H^T \Phi^T B \end{bmatrix} f + \begin{bmatrix} (M_B + \hat{\Phi}_2 M \hat{\Phi}_1) \alpha_B + \hat{\Phi}_2 M \Phi H \ddot{q} \\ H^T \Phi^T M \hat{\Phi}_1 \alpha_B + H^T \Phi^T M \Phi H \ddot{q} \end{bmatrix} + \begin{bmatrix} \hat{\Phi}_2 M \Phi a + b_B + \hat{\Phi}_2 b \\ H^T \Phi^T M \Phi a + H^T \Phi^T b \end{bmatrix}, \quad (\text{A.16})$$

where

$$\hat{\Phi}_1 = \begin{bmatrix} \Phi_{1B} \\ \vdots \\ \Phi_{nB} \end{bmatrix}, \quad (\text{A.17})$$

and

$$\hat{\Phi}_2 = [\Phi_{1B}^T \quad \dots \quad \Phi_{nB}^T]. \quad (\text{A.18})$$

Finally, we obtain

$$\begin{bmatrix} f_b \\ \tau \end{bmatrix} = \begin{bmatrix} M_b & M_{rb} \\ M_{br} & M_r \end{bmatrix} \begin{bmatrix} \alpha_B \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} C_B \\ C_r \end{bmatrix} + \begin{bmatrix} \Phi_{n+1,B}^T \\ J^T \end{bmatrix} f, \quad (\text{A.19})$$

where

$$\begin{aligned}M_b &= M_B + \hat{\Phi}_2 M \hat{\Phi}_1, \\ M_{rb} &= \hat{\Phi}_2 M \Phi H, \\ M_{br} &= H^T \Phi^T M \hat{\Phi}_1, \\ M_r &= H^T \Phi^T M \Phi H, \\ C_B &= \hat{\Phi}_2 M \Phi a + b_B + \hat{\Phi}_2 b, \\ C_r &= H^T \Phi^T M \Phi a + H^T \Phi^T b.\end{aligned}$$

Note that the well-known manipulator Jacobian is given by $J = H^T \Phi^T B$. Similarly, M_r and C_r are the mass-inertia matrix and Centrifugal/Coriolis terms obtained for a standard fixed-base manipulator arm (Jain, 2010).