Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research xx (2022) xxx-xxx

ADVANCES IN SPACE RESEARCH
(a COSPAR publication)

www.elsevier.com/locate/asr

Post-Capture Detumble Trajectory Stabilization for Robotic Active Debris Removal

Shubham Vyas[a,*], Lasse Maywald[a], Shivesh Kumar[a], Marko Jankovic[a], Andreas Mueller[b], Frank Kirchner[a,c]

[a]Robotics Innovation Center, DFKI GmbH, Robert-Hooke-Strasse 1, Bremen 28359, Germany
[b]Institute of Robotics, Johannes Kepler University, Linz 4040, Austria
[c]Robotics Research Group, University of Bremen, Bremen 28359, Germany

Abstract

Recent increase in space debris combined with the increase in the number of satellites launched has created an increased risk of collisions. The effects of the increased risk can be seen in the form of an increased number of *near misses* in recent years. The use of robotic manipulators has been suggested for Active Debris Removal (ADR) to reduce the risk of potential future collisions that generate more debris in the orbits around Earth. Compared to other ADR methods, robotic manipulators provide increased versatility as they can be reused for On-Orbit Servicing as well as On-Orbit Assembly missions. A robotic ADR operation consists of three phases: Approach, Capture, and Detumble. This paper provides a method for performing feedback-based stabilization of post-capture detumble trajectories of the chaser-debris system. The approach presented here uses Time-Varying Linear Quadratic Regulator (TVLQR) for stabilization along the detumble trajectory. The contributions of this paper are as follows: A quaternion-based linearization method for multibody systems with a free-floating base, TVLQR for stabilizing the optimal detumble trajectory, and a probabilistic Region of Attraction analysis of the resulting closed-loop system. The estimated Region of Attraction could serve as the goal for the capture controller thus allowing for controller composition through ADR phases while guaranteeing stability and successful detumble.

Keywords: Active Debris Removal ; Space Robotics ; Trajectory Stabilization

1. Introduction

An increase in space debris in recent times has led to an increased risk of collisions between debris objects and functional satellites (ESA Space Debris Office, 2022; Anz-Meador, 2020). This growth in debris population has accentuated the need for Active Debris Removal (ADR) (Liou, 2011). Robotic manipulators have also been suggested as one of the methods for ADR. Along with ADR, their applicability for On-Orbit Servicing (OOS) and On-Orbit Assembly (Graham et al., 1979) demonstrates their versatility. Thus, the technologies developed for control and planning for Robotic ADR could be applied to these other applications as well. A robotic ADR mission consists of 3 phases during the proximity operations: Approach, Capture and Detumble. These are illustrated in Figure 1. A robotic ADR mission requires a successful execution of all 3

*Corresponding author.
Email addresses: Shuhbam.Vyas@dfki.de (Shubham Vyas), lasse_jenning.maywald@dfki.de (Lasse Maywald), Shivesh.Kumar@dfki.de (Shivesh Kumar), Marko.Jankovic@dfki.de (Marko Jankovic), a.mueller@jku.at (Andreas Mueller), Frank.Kirchner.dfki.de (Frank Kirchner)

https://dx.doi.org/10.1016/j.jasr.xxxx.xx.xxx

phases. The success of a phase depends on how well the previous phase was executed. For example, an imperfect capture which involves large contact forces might result in a system state which cannot be stabilized or detumbled using the available control resources. Similarly, an approach which is out of certain bounds could result in a foiled capture. This highlights the need for the analysis of the controllers used in each phase to understand their bounds, or Regions of Attraction (RoA). Any initial state within the RoA can then be guaranteed to be driven to the respective phase's goal. This RoA could then be goal for the preceding phase's controller thereby connecting the phases while guaranteeing a successful robotic ADR mission. Similar ideas of Sequential Controller Composition have been applied in other fields of robotics to guarantee stability between phase and controller transitions (Burridge et al., 1999).

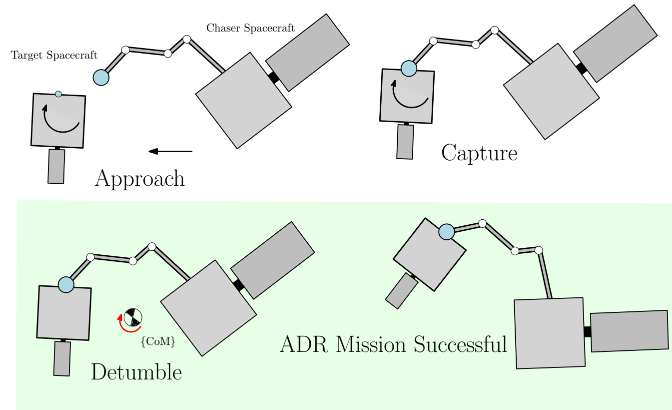


Figure 1: Various Phases in robotic ADR proximity operations. The top left figure shows the Approach phase with the Chaser spacecraft with the robot arm approaching the target from the right. The focus of this paper are the bottom phases. Bottom left figure illustrates an ideal Post-Capture Detumble Scenario with Perfect Synchronisation between Chaser's end-effector and Target's grasping point. The combined chaser-target system then rotate about the system's Center of Mass (CoM).

In this work, we focus on the detumble phase of robotic ADR. Other non-robotic detumble methods have been presented in the literature (Mark & Kamath, 2019), such as by using lasers (Vetrisano et al., 2015) or Eddy currents (Gómez & Walker, 2015). These non-contact methods, while applicable for certain debris types (Jankovic et al., 2020), are not without their challenges. In this work, we focus on detumble using a robotic manipulator as the methods presented here can be extended to On-Orbit Servicing (OOS) and On-Orbit Assembly in the future. Space robots differ from their traditional robotic counterparts on Earth as they have a floating base in orbital environment in contrast to the fixed base robots operating under gravity on Earth. Due to this, the base spacecraft is free to move under the influence of the reaction wrenches generated during the operation of the robotic manipulator. This leads to a coupling of kinematics and dynamics and requires kinodynamic planning and control even for the simplest of tasks (Papadopoulos & Dubowsky, 1991; Dubowsky & Papadopoulos, 1993; Flores-Abad et al., 2017). For the method presented in this paper, we compute time and effort optimal detumble trajectory of the full post-capture system which takes into account the kino-dynamic coupling, while satisfying the actuation limit constraints. We define an ideal capture scenario and use that to find the initial conditions for the detumble trajectory optimization. We then use the dynamics as well as the joint torque limits as constraints along with costs on the control inputs, final velocity, and time. The trajectory optimization method provided in this work is kept generic to allow for general path or boundary constraints, and additional costs a trajectory optimization might need to include for a future mission. Once the feasible and optimal trajectory is obtained, we utilize a Time Varying Linear Quadratic Regulator (TVLQR) as a stabilizing controller to execute the trajectory online. Since the trajectory optimization uses quaternions for Chaser spacecraft's attitude representation, we derive a quaternion-based linearization of the free-floating multibody system dynamics and utilize this new linearization method for synthesizing the TVLQR. We validate this controller using a full dynamic simulation of the system using Drake (Tedrake & the Drake Development Team, 2019) software framework. To find the robustness limits of the derived TVLQR controller, we perform a probabilistic Lyapunov-based RoA analysis which provides a probabilistic guarantee for the region of state-space that this controller can stabilize and drive towards the goal. This results in a controller for detumbling trajectory stabilization which is robust to disturbances and is also certifiable for disturbances in the detumble trajectory which the controller can recover from. This RoA can then be used by the preceding phase's capture controller as a goal to ensure controller composition through ADR phases.

This paper is organized as follows, Section 2 provides an overview of the related works, which is followed by an introduction to free-floating kinematics and dynamics in Section 3. The trajectory optimization is detailed in Section 4 followed by a method for quaternion-based linearization and trajectory stabilization in Section 5. The RoA analysis of the TVLQR controller is provided in Section 6. Section 7 concludes this work.

2. Related Works

Various control strategies have been developed for kinodynamic planning and control of free-floating orbital robots. Early works on the control of space robots with free-floating base focused on obtaining good end-effector trajectory tracking performance while taking into account the free-floating base. To achieve this, various approaches such as the virtual manipulator (Vafa & Dubowsky, 1990; Dubowsky & Papadopoulos, 1993), disturbance map (Dubowsky & Papadopoulos, 1993), and generalized Jacobian matrix (Umetani & Yoshida, 1989) can be found in the literature. These approaches provide methods to find the joint control inputs to track a given end-effector trajectory while accounting for the free-floating base. A more comprehensive review of end-effector trajectory tracking and control methods can be found in Flores-Abad et al. (2017). Control of free floating robotic systems in other application areas such as underwater robotics also focus on end-effector trajectory tracking (Hildebrandt et al., 2008). However, the source and purpose of the trajectory is not taken into account in the methods given above. This can be applicable for tasks such as OOS and On-Orbit Assembly where the end-effector trajectory planning problem can be solved using the higher-level problem constraints beforehand and this trajectory can then be tracked to accomplish the task. However, the path planning problem and control problem are intertwined for ADR as the path required to be followed cannot be fully determined previously and is emergent from the state and dynamic properties of the system, primarily the target, during operation. This has prompted the development of optimization-based methods for planning and control. Here, the higher-level goals for ADR are provided to the optimization solver along with the system's dynamics as constraints. Some of the most common goals for optimization-based methods is the minimization of time and control effort. The solver then provides a state and control input trajectory which is consistent with the dynamics, satisfies the given constraints, accomplishes the given goals, along with minimizing the given costs. One of the earliest works in which the trajectory was derived from a higher-level goal are the *Reaction Null-Space* control and *Bias-Momentum approach*. Nenchev & Yoshida (1999) provide an impact model and post-impact control using inverse dynamics Proportional-Derivative (PD) control for damping joint motion post-impact along with reaction null-space control for keeping the base attitude unchanged. Dimitrov & Yoshida (2004a,b) pre-load the chaser spacecraft's arm with target's angular momentum during approach to detumble without affecting the attitude of the chaser. This is known as the *Bias-momentum* approach. They further use the reaction null-space and angular momentum equation to derive joint torque control law for the post-impact/contact phase of the mission. Aghili (2013, 2009b,c, 2010, 2020) derives the torque required for time-optimal detumble of the target while taking into account the maximum torque applied on the target by the end-effector of the chaser's robot arm. The end-effector torque is then controlled using feedback linearization. They also use PD torque control for base attitude. Even though end-effector applied torque limits are considered, joint torque limits and other constraints are not included. This approach is then extended in Aghili (2008, 2009a) to include the approach phase by synchronizing the end-effector velocity during approach to the velocity of the grasping point on the target. In Shibli et al. (2006), inverse dynamics based control is carried out with contact constraints. However, joint/base torque limits are not considered. In Matunaga et al. (2001), the detumbling is carried out using cushion-type damper attached to the end-effector of the chaser's robot arm during contact using slide/push-based method. In their work, the capture post detumble is not discussed. Rybus et al. (2014) provide a Rapidly-exploring Random Trees based trajectory planner to minimize rotational kinetic energy of the system post joint rigidization and stabilize the motion about one axis. Furthermore, Rybus et al. (2016) compare the optimal motion generated using trajectory optimization to a reference straight line trajectory. The optimal trajectory shows substantially lower power usage for the joint motors. They predict that the torque demands for detumbling might be higher and hence optimization is an useful approach. Zhang et al. (2017) use an adaptive sliding mode control for detumble of a large inertia target. No torque limits on base/joints are considered. In Wang et al. (2018b), the end-effector detumble trajectory generation was carried out using quartic Bézier curves and an adaptive differential evolution algorithm while taking into account constraints such as end-effector torque (but no joint torque) and target motion. The optimal end-effector path minimizes time and control effort. Feedback linearization is then used to compute the system's control efforts for the given end-effector trajectory and PD control is used for joint/base torques. Similarly, (Wang et al., 2018a) also use quartic Bézier curves and particle swarm optimization to find optimal end-effector trajectory for detumbling. They also then PD control for joint/base torques. In more recent work, Raina et al. (2021) model the system dynamics for approach, impact, and post-impact for capture using a dual-arm chaser satellite. The pre-impact trajectory tracking uses PD control and the impact dynamics are estimated using the derived impact model. Post-impact reaction null-space control is performed using the initial conditions from the given impact modeling. No joint constraints were considered.

While few of the above mentioned works consider other phases (approach/capture) during the planning and control for detumbling, none of these studies provide any robustness analysis for phase transitions or due to the effects of contact during capture. Furthermore, few of the methods consider the actuation limits of the chaser spacecraft and its robot arm. This can have a large effect on the detumble trajectory as the control capabilities of the system might affect the optimal trajectory that can be followed. The trajectory optimization methods given above generally consider the optimal trajectory of the target body with the constraint on the torques applied on the target. They do not consider the full dynamic trajectory of the combined chaser-target system during the optimization procedure. This end-effector trajectory is then usually followed using an inverse dynamics PD control. However, such methods do not take into account the state/actuation limits of the full system such as the position, velocity, or torque limits of the joints and base. To the best of author's knowledge, a full system (combined chaser-target) trajectory optimization along with trajectory stabilization and the controller's region of attraction analysis have not yet been reported in this field. These results would allow for better controller composition during a robotic ADR mission. The salient contributions of this paper are as follows:

- A generic non-linear post-capture detumble trajectory optimization formulation capable of including joint, base, and end-effector wrench limits. The method is also easily extendable to include other constraints such as collision constraints.
- A quaternion-based linearization of free-floating multibody system which allows for model-based control for trajectory tracking.
- A linear trajectory tracking controller (TVLQR) which can track any feasible detumble trajectory.
- Region of Attraction analysis of the tracking controller which can be used as a goal for the capture controller to guarantee successful detumble.

We approach the detumble problem as a two step process in this paper. First a non-linear trajectory optimization is carried out which takes in account the system-level constraints such as position, velocity, and actuation limits. We then use TVLQR as a trajectory stabilization controller to execute the given trajectory. The robustness of the controller is evaluated using RoA analysis. Numerical simulations are carried out to validate the given methods. Supplementary Video of the simulations can be found at: `https://youtu.be/wXCkwEe7IL0`.

3. Modeling of Floating-Base Systems

3.1. Kinematics and Dynamics Motion Equations

The dynamics modeling of floating base systems has been a research topic since 1980's, and there are various approaches tailored to the specific application needs. Notably the centroidal dynamics formulation (Dubowsky & Papadopoulos, 1993; Papadopoulos & Dubowsky, 1993a) has gained popularity for control of humanoids and legged robots (Orin & Goswami, 2008; Orin et al., 2013). An important aspect of the kinematics and dynamics formulation is that it requires minimal effort in terms of parameter conventions, reusability, and modularity. In this respect, the Lie group and screw formulations (Park et al., 1995; Lynch & Park, 2017; Müller, 2018b,a) are clearly the method of choice. Moreover, its consistent mathematical setting makes ideal for numerical simulation (using Lie-group integration schemes) and further offer insight into the geometry of motion. Such a formulation is briefly summarized in this section.

A space-fixed frame $\mathcal{F}_0$ is introduced, and a frame $\mathcal{F}_\mathrm{b}$ is attached at the moving base (see Figure 2). The configuration of the floating-base system is described by the configuration of the base relative to $\mathcal{F}_0$ and by the pose of the attached kinematic chains relative to the base. The latter is also referred to as the 'shape'. Denote with $\vartheta \in \mathbb{V}^n$ the joint coordinate vector of the kinematic chain, and with $\mathbf{C}_\mathrm{b} \in SE(3)$ the transformation matrix from $\mathcal{F}_\mathrm{b}$ to $\mathcal{F}_0$. The configuration is then described by $\mathbf{q} = (\mathbf{C}_\mathrm{b}, \vartheta) \in SE(3) \times \mathbb{V}^n$, and $SE(3) \times \mathbb{V}^n$ is regarded as configuration space. The corresponding generalized velocity is introduced $\mathbf{s} := (\mathbf{V}_\mathrm{b}, \dot{\vartheta}) \in \mathbb{R}^{6+n} \cong se(3) \times \mathbb{R}^n$ where $\mathbf{V}_\mathrm{b} = (\omega_\mathrm{b}, \mathbf{v}_\mathrm{b}) \in \mathbb{R}^6 \cong se(3)$ is the twist of the base body, which will be in body-fixed representation.

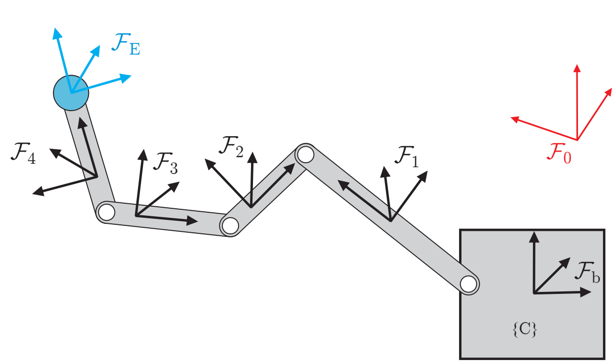


Figure 2: Assignment of body-fixed frames. $\mathcal{F}_0$ denotes the inertial frame, $\mathcal{F}_\mathrm{b}$ the frame at the base of the chaser, and $\mathcal{F}_\mathrm{i}$ the frame at link i of the robot arm.

A body-fixed frame $\mathcal{F}_i, i = 1, \ldots, n$ is attached at each link of the tree-topology system. The pose of body i relative to the base frame $\mathcal{F}_\mathrm{b}$ is determined by the product of exponentials (Brockett, 1984; Lynch & Park, 2017)

$$\mathbf{C}_i(\vartheta) = \exp(\mathbf{Y}_1 \vartheta_1) \exp(\mathbf{Y}_2 \vartheta_2) \cdot \ldots \cdot \exp(\mathbf{Y}_i \vartheta_i) \mathbf{A}_i. \tag{1}$$

where $\mathbf{Y}_j \in \mathbb{R}^6$ is the screw coordinate vector of joint j in the reference configuration $\vartheta = \mathbf{0}$ represented in the base frame, and $\mathbf{A}_i = \mathbf{C}_i(\mathbf{0})$ is the reference configuration of body i relative to the base frame. Denote with $\bar{\mathbf{V}}_i$ the absolute twist of body i relative to $\mathcal{F}_0$ represented in $\mathcal{F}_i$, and with $\mathsf{V}_\mathrm{b} = \left(\mathbf{V}_\mathrm{b}, \bar{\mathbf{V}}_1, \ldots, \bar{\mathbf{V}}_n\right)$ the body-fixed system twist. Further, denote with $\mathbf{X}_i \in \mathbb{R}^6$ the constant screw coordinate vector of joint i represented in $\mathcal{F}_i$. The system twist is expressed in terms of the generalized state as

$$\mathsf{V}_\mathrm{b} = \mathsf{J}_\mathrm{b} \mathbf{s} \tag{2}$$

where the geometric system Jacobian $\mathsf{J}_\mathrm{b}(\vartheta)$ is given by

$$\mathsf{J}_\mathrm{b} = \mathsf{A}_\mathrm{b} \mathsf{X}_\mathrm{b}, \text{ with } \mathsf{A}_\mathrm{b} := \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathsf{U} & \mathsf{A} \end{pmatrix}, \mathsf{X}_\mathrm{b} := \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathsf{X} \end{pmatrix} \tag{3}$$

and with $\mathsf{A}(\vartheta) \in \mathbb{R}^{6 \times n, 6 \times n}$, $\mathsf{U}(\vartheta) \in \mathbb{R}^{6 \times n, 6}$, and $\mathsf{X} \in \mathbb{R}^{6 \times n, n}$ defined as

$$\mathsf{A} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ & & & \ddots & \vdots \\ & \mathbf{Ad}_{\mathbf{C}_{i,j}} & & & \mathbf{0} \\ & & & & \mathbf{I} \end{pmatrix}, \mathsf{U} = \begin{pmatrix} \mathbf{Ad}_{\mathbf{C}_1}^{-1} \\ \mathbf{Ad}_{\mathbf{C}_2}^{-1} \\ \vdots \\ \mathbf{Ad}_{\mathbf{C}_n}^{-1} \end{pmatrix} \tag{4}$$

$$\mathsf{X} = \operatorname{diag}\left(\mathbf{X}_1, \ldots, \mathbf{X}_n\right). \tag{5}$$

The equations of motion (EOM) of the floating-base system are written in the form

$$\mathbf{M}(\vartheta) \dot{\mathbf{s}} + \mathbf{C}(\vartheta, \mathbf{s}) \mathbf{s} + \mathbf{Q}^{\mathrm{grav}}(\mathbf{q}) = \mathbf{Q}^{\mathrm{act}} \tag{6}$$

where $\mathbf{Q}^{\mathrm{act}} = (\mathbf{W}^{\mathrm{act}}, \tau)^T \in \mathbb{R}^{6 \times n}$ consists of the wrench $\mathbf{W}^{\mathrm{act}} \in \mathbb{R}^6$ acting at the base, and the vector of joint control torques

$\tau \in \mathbb{R}^n$. $\mathbf{Q}^{\mathrm{grav}}(\mathbf{q})$ is the vector of generalized gravity forces which are taken to be zero for orbital free-floating robots. The base wrench $\mathbf{W}^{\mathrm{act}}$ is used to model the effect of thrusters or reaction-wheels.

All terms in Equation 6 can be expressed in closed form in terms of simple algebraic operations. To this end, denote with $\mathbf{M}_{\mathrm{b}}$ the mass matrix of the base w.r.t. $\mathcal{F}_{\mathrm{b}}$, and with $\mathbf{M}_i, i = 1,\ldots,n$ the mass matrix of link i w.r.t. $\mathcal{F}_i$. Then the mass and Coriolis matrix are

$$\mathbf{M}(\vartheta) = \mathsf{J}_{\mathrm{b}}^T \mathsf{M}_{\mathrm{b}} \mathsf{J}_{\mathrm{b}} \tag{7}$$

$$\mathbf{C}(\vartheta, \mathbf{s}) = \mathsf{J}_{\mathrm{b}}^T (\mathsf{M}_{\mathrm{b}} \mathsf{A}_{\mathrm{b}} \mathsf{b}_{\mathrm{b}} \mathsf{X}_{\mathrm{b}} - \mathsf{b}_{\mathrm{b}}^T \mathsf{M}_{\mathrm{b}} \mathsf{J}_{\mathrm{b}}). \tag{8}$$

where

$$\mathsf{M}_{\mathrm{b}} = \mathrm{diag}(\mathbf{M}_{\mathrm{b}}, \mathbf{M}_1, \ldots, \mathbf{M}_n) \tag{9}$$

$$\mathsf{b}_{\mathrm{b}}(\mathsf{V}_{\mathrm{b}}) = \mathrm{diag}\,(\mathbf{ad}_{\mathbf{V}_{\mathrm{b}}}, \mathbf{ad}_{\bar{\mathbf{V}}_1}, \ldots, \mathbf{ad}_{\bar{\mathbf{V}}_n}) \tag{10}$$

and $\mathbf{ad}$ is the 6×6 matrix corresponding to the adjoint action, i.e. the Lie bracket on $se(3)$, so that $\mathbf{ad}_{\mathbf{Y}}\mathbf{X}$ is the screw product of two screw coordinate vectors (Lynch & Park, 2017; Müller, 2018b). The vector of generalized actuation forces is given with the system Jacobian as $\mathbf{Q}^{\mathrm{act}} = \mathsf{J}_{\mathrm{b}}^T \mathsf{W}_{\mathrm{b}}^{\mathrm{act}}$.

Remark. The above closed form expressions can be readily translated into recursive $O(n)$ algorithms for solving the equations of motion (Jain, 2011; Featherstone, 2008). In particular, Recursive Newton Euler Algorithm (RNEA) is typically used to solve the inverse dynamics of robots and the Articulated Body Algorithm (ABA) is used for solving the forward dynamics in a computationally efficient manner.

3.2. Momentum Conservation and Decoupling

Various aspects of control and simulation of space robots are to be noticed. The EOM (Equation 6) can be separated for base and arm dynamics. The base dynamics is governed by the momentum balance (the Euler-Poincaré equations) and the manipulator dynamics by the Lagrange equations. The mass matrix can be split accordingly as

$$\mathbf{M}(\vartheta) = \begin{pmatrix} \mathbf{M}_{\mathrm{bb}} & \mathbf{M}_{\mathrm{bc}} \\ \mathbf{M}_{\mathrm{bc}}^T & \mathbf{M}_{\mathrm{cc}} \end{pmatrix} \tag{11}$$

which allows to separate the base momentum

$$\Pi_{\mathrm{b}}(\vartheta, \mathbf{s}) = \mathbf{M}_{\mathrm{bb}}(\vartheta)\mathbf{V}_{\mathrm{b}} + \mathbf{M}_{\mathrm{bc}}(\vartheta)\dot{\vartheta}. \tag{12}$$

$\mathbf{M}_{\mathrm{bb}}$ is called the locked inertia tensor. Assuming zero initial base momentum, i.e. $\Pi_{\mathrm{b}}(\vartheta, \mathbf{s}) = \mathbf{0}$, the momentum conservation yields the relation

$$\mathbf{V}_{\mathrm{b}} = \mathbf{F}_{\vartheta}\dot{\vartheta}, \text{ with } \mathbf{F}_{\vartheta}(\vartheta) := -\mathbf{M}_{\mathrm{bb}}^{-1}\mathbf{M}_{\mathrm{bc}} \tag{13}$$

where $\mathbf{M}_{\mathrm{bb}}^{-1}\mathbf{M}_{\mathrm{bc}}$ is called dynamic coupling factor (Xu, 1993). Similarly, the Coriolis matrix can be split into the floating base and manipulator parts according to

$$\mathbf{C}(\vartheta, \mathbf{s}) = \begin{pmatrix} \mathbf{C}_{\mathrm{bb}} & \mathbf{C}_{\mathrm{bc}} \\ \mathbf{C}_{\mathrm{bc}}^T & \mathbf{C}_{\mathrm{cc}} \end{pmatrix}. \tag{14}$$

Substituting Equation 13 into Equation 6 while assuming $\mathbf{W}^{\mathrm{act}} = \mathbf{0}$ yields the reduced EOM in terms of the joint variables

$$\mathbf{M}_{\vartheta}\ddot{\vartheta} + \mathbf{C}_{\vartheta}\dot{\vartheta} = \tau \tag{15}$$

with the reduced system inertia matrix and the reduced system Coriolis matrix

$$\mathbf{M}_{\vartheta}(\vartheta) := \mathbf{M}_{\mathrm{cc}} + \mathbf{M}_{\mathrm{bc}}^T \mathbf{F}_{\vartheta} \tag{16}$$

$$\mathbf{C}_{\vartheta}(\vartheta, \dot{\vartheta}) := \mathbf{C}_{\mathrm{cc}} + \mathbf{F}_{\vartheta}^T \mathbf{C}_{\mathrm{bb}} \mathbf{F}_{\vartheta} + \mathbf{F}_{\vartheta}^T \mathbf{C}_{\mathrm{bc}} + \mathbf{C}_{\mathrm{cb}} \mathbf{F}_{\vartheta}. \tag{17}$$

These EOM are the basis for controlling the robot. The resulting base motion is determined by the solution of Equation 13. This also gives rise to an attitude control problem. As the momentum of the entire system is conserved, it is to be noticed that the momentum conservation represents a non-holonomic constraint, and the control problem in terms of ϑ is non-holonomic. The Equation 13 serve as kinematic reconstructions that determine the attitude from the motion of the arm.

3.3. Forward kinematics

The EE (gripper) is attached at the last link n of the robotic arm of a space robot, and is represented by an EE-frame $\mathcal{F}_{\mathrm{E}}$. The EE-twist relative to the base is then determined as $\mathbf{V}_{\mathrm{E}} = \mathbf{J}_{\mathrm{E}}\dot{\vartheta}$. The latter is the last block row of J_{b} in Equation 3, and can thus be computed efficiently. With the relation in Equation 13, the EE-twist relative the to the inertial frame $\mathcal{F}_0$ is determined as

$$\bar{\mathbf{V}}_{\mathrm{E}} = \bar{\mathbf{J}}_{\mathrm{E}}\dot{\vartheta} \tag{18}$$

where

$$\bar{\mathbf{J}}_{\mathrm{E}} = \mathbf{Ad}_{\mathbf{C}_{\mathrm{E}}}^{-1}\mathbf{J}_{\mathrm{E}} + \mathbf{F}_{\vartheta} \tag{19}$$

is the *generalized EE-Jacobian* (Umetani & Yoshida, 1987, 1989; Yoshida & Umetani, 1993; Yoshida & Nenchev, 1998), and $\mathbf{C}_{\mathrm{E}}$ is the pose of the last link determined by Equation 1.

In addition to the well-known kinematic singularities, the generalized Jacobian exhibits special singularities. Configurations where the rank of $\bar{\mathbf{J}}_{\mathrm{E}}$ drops are called *dynamics singularities* (Papadopoulos & Dubowsky, 1991, 1993b). Consequences of dynamic singularities for the control of space robots, and the determination of singularity free workspace, are discussed in Papadopoulos & Dubowsky (1991, 1993b,a); Nanos & Papadopoulos (2012).

4. Trajectory Optimization

In this section, we describe the trajectory optimization method for finding a post-capture detumble trajectory. For the post-capture detumble scenario, we consider the case of a chaser spacecraft with a 3 Degree of Freedom (DoF) robot arm capturing a tumbling target satellite. We consider a target tumbling at a rate of $5\,^{\circ}\,\mathrm{s}^{-1}$ about a given axis. The chaser spacecraft has a mass of 100 kg (evenly distributed about a cube of a 2 m side), the links have masses of 10 kg, 8 kg, and 4 kg and lengths of 0.9 m, 0.7 m, and 0.3 m. The target spacecraft has a mass of 50 kg (evenly distributed about a cube of a 0.6 m side).

4.1. Capture Scenario

Here, we describe an ideal debris capture during a robotic ADR mission. Although the ideal capture is not achievable during operation, its description provides properties of the system during the ADR operation which are used in the following sections of this paper. An ideal capture is obtained when the end-effector of the robotic arm on the chaser spacecraft is perfectly in sync with the grasp point on the target. This implies that the relative velocity between the end-effector and the grasp point is zero. Once such state is obtained, we assume that the chaser then captures the target without applying any contact forces on the target due to the zero relative velocity between the contact surfaces. Post-contact, the grasping mechanism locks the system such that a rigid connection can be assumed between the chaser-target system. The target then acts as an extension to the end-effector link in terms of the link's mass-inertia properties within the chaser-target multi-body system. Post-capture, the state of the system is thus identical to the pre-capture synchronized state with the only difference being that the chaser and target spacecraft are now connected via a rigid connection and thus can be assumed as a singular larger multi-body system. The generalized velocities of the combined system post-capture are given by the generalized velocities of the chaser as the target is now an extension of the chaser's rigid body chain. The realization of the chaser spacecraft's initial syncing maneuver, i.e. approach, is assumed to be possible and its trajectory optimization and stabilization are not considered in the focus of this paper. The post-capture state described here is used as the initial condition for the trajectory optimization described in the following section. For an ideal capture, a zero-wrench contact map is thus assumed resulting in no change in velocities pre- and post-capture.

4.2. Initial State for Detumbling

To find the initial state of the system the capture scenario mentioned in Sub-Section 4.1 is used. Post-capture, the combined chaser-target system's initial state should be such that the rotation of the target is maintained (due to perfect synchronization). For this, Resolved Motion Rate Control (RMRC) with the Generalized Jacobian Matrix (GJM) (Umetani & Yoshida, 1989) are used to find the initial state of the combined chaser-target system. The initial state was estimated such that the end-effector (in this case, the Center of Mass (CoM) of the target object) has the given initial rotation rate. RMRC is used to find the chaser robot arm's joint rates at a feasible capture configuration which results in perfect synchronization. The GJM of the target, denoted by $\bar{\mathbf{J}}_{\mathrm{t}}$ is given in Equation 19. The RMRC can be now expressed as:

$$\dot{\vartheta} = \bar{\mathbf{J}}_{\mathrm{t}}^{-1}\bar{\mathbf{V}}_{t} \tag{20}$$

Here, $\dot{\vartheta} \in \mathbb{R}^n$ are the joint rates and $\bar{\mathbf{V}}_t \in \mathbb{R}^6$ represents the twist vector of the target spacecraft. Equation 20 can be used to determine the joint rates required for perfect synchronization with the rotation rate of the target. This provides the system state post-capture.

4.3. Problem Formulation

In this section, the trajectory optimization problem is formulated. Then, the initial and final states can be used to transcribe it into a Non-Linear Programming (NLP) problem and solve it using off the shelf NLP solvers. To solve the trajectory optimization problem, the direct collocation (Kelly, 2017; Betts, 2010) method was used. The optimization problem formulation relies on minimizing the following costs: detumble time, actuator effort, and final velocities while satisfying the following constraints: system dynamics, initial state and actuation limits. The costs and constraints for the trajectory optimization problem can be written as follows:

$$\min_{\mathbf{q},\mathbf{Q}^{\text{act}}} \int_0^{t_f} (w_t \Delta t + \mathbf{Q}^{\text{act}^T} \mathbf{w}_{\text{act}} \mathbf{Q}^{\text{act}}) dt + \mathbf{s}_f^T \mathbf{w_s} \mathbf{s}_f \tag{21a}$$

$$subject\ to:$$

$$\mathbf{M}(\vartheta)\dot{\mathbf{s}} + \mathbf{C}(\vartheta, \mathbf{s})\mathbf{s} = \mathbf{Q}^{\text{act}}(t) \tag{21b}$$

$$\dot{\mathbf{q}}_{\mathbf{b}} = \frac{1}{2}\Omega(\omega_{\mathbf{b}})\mathbf{q}_{\mathbf{b}} \tag{21c}$$

$$\|\mathbf{q}_{\mathbf{b}}\| = 1 \tag{21d}$$

$$\mathbf{q} \subset \mathbf{q}_{lim}, \ \ \mathbf{s} \subset \mathbf{s}_{lim} \tag{21e}$$

$$\mathbf{Q}^{\text{act}} \subset \mathbf{Q}^{\text{act}}_{lim} \tag{21f}$$

$$\mathbf{q}(t_0) = \mathbf{q}_0 \tag{21g}$$

Here, $w_t, \mathbf{w}_{\text{act}}$, and $\mathbf{w_s}$ are the weights for the time, actuator effort, and final velocity costs respectively. Note that the time and actuator costs are cumulative costs whereas the final velocity cost is a terminal cost. Equations 21b-21g represent the constraints used for optimization: dynamics, state limits, actuation limits, and initial condition respectively. The dynamics constraint in Equation 21b represents the equations of motion for a multi-body system without gravity from Section 3. For the trajectory optimization, the base-orientation of the system configuration $\mathbf{q}$ is parameterised using a unit-quaternion $\mathbf{q_b}$. Equation 21c gives the mapping between angular velocity and the quaternion derivatives i.e. quaternion kinematics for the floating base of the chaser spacecraft (Andrle & Crassidis, 2013). $\mathbf{q}$ represents the generalized coordinates of the system whereas $\mathbf{q_b}$ is the rotation representation of the chaser's base using a unit quaternion in Equation 21d.

4.4. Transcription and Results

The trajectory optimization formulation given in Sub-Section 4.3 is discretized using Euler's method (first order approximation) and solved using the *SNOPT (Sparse Nonlinear OPTimizer)* solver (Gill et al., 2005). Higher order methods for trajectory optimization can be used to obtain better interpolation results (Betts, 2010; Jankovic & Kirchner, 2018). For the purpose of this work, Euler's method provides a sufficiently accurate model for stabilization and satisfies the dynamics constraints at the knot points. $N = 100$ knot/collocation points were considered for the discretised optimization problem. The time-step for the trajectory is taken as an optimization variable and is constrained between 0.01 s and 0.2 s. This allows us to formulate the detumbling as an optimal-time problem. The discretized equations for the *SNOPT* solver can be written as follows:

$$\min_{\mathbf{q}[k],\mathbf{Q}^{\text{act}}} \sum_{k=1}^{N} \Delta t[k](w_t + \mathbf{Q}^{\text{act}}[k]^T \mathbf{w}_{\text{act}} \mathbf{Q}^{\text{act}}[k]) + \mathbf{s}[N]^T \mathbf{w_s} \mathbf{s}[N] \tag{22a}$$

$$subject\ to:$$

$$\mathbf{M}[k+1]\dot{\mathbf{s}}[k] + +\mathbf{C}[k+1] - \mathbf{Q}^{\text{act}}[k] = 0, \ \forall k \subset [0, N-1] \tag{22b}$$

$$\mathbf{s}[k+1] = \mathbf{s}[k] + \Delta t[k]\dot{\mathbf{s}}[k], \ \forall k \subset [0, N-1] \tag{22c}$$

$$\mathbf{q}[k+1] = \mathbf{q}[k] + \Delta t[k]\mathbf{s}[k+1], \ \forall k \subset [0, N-1] \tag{22d}$$

$$\dot{\mathbf{q}}_{\mathbf{b}}[k+1] = \frac{1}{2}\Omega(\omega_b[k+1])\mathbf{q}_{\mathbf{b}}[k], \ \forall k \subset [0, N-1] \tag{22e}$$

$$\|\mathbf{q}_{\mathbf{b}}[[k]\| = 1, \ \forall k \subset [0, N] \tag{22f}$$

$$\Delta t[k+1] = \Delta t[k], \ \forall k \subset [0, N-1] \tag{22g}$$

$$\mathbf{Q}^{\text{act}}[k] \subset \mathbf{Q}^{\text{act}}_{lim}, \ \forall k \subset [0, N] \tag{22h}$$

$$\mathbf{q}[k] \subset \mathbf{q}_{lim}, \ \forall k \subset [0, N] \tag{22i}$$

$$\mathbf{s}[k] \subset \mathbf{s}_{lim}, \ \forall k \subset [0, N] \tag{22j}$$

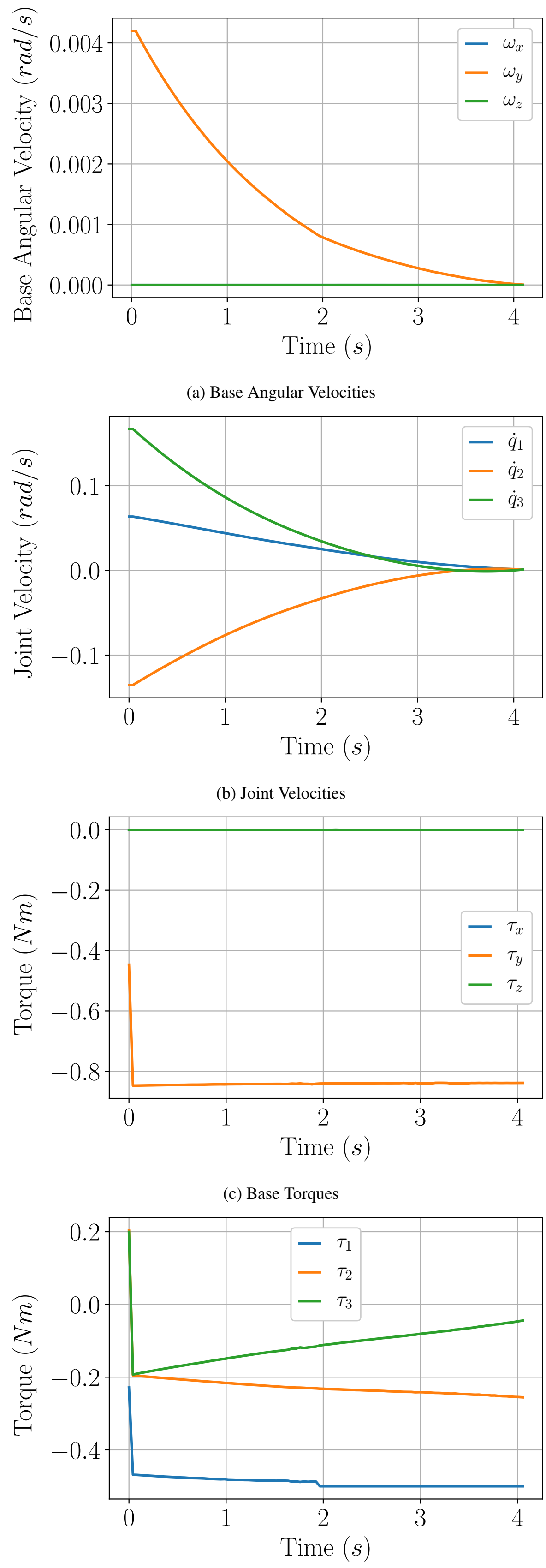
$$\mathbf{q}[0] = \mathbf{q}_0 \tag{22k}$$

$$\mathbf{s}[0] = \mathbf{s}_0 \tag{22l}$$

The above given problem and constraints were solved using the *SNOPT* solver in Drake. The constraints in Equation 22 were initially relaxed to obtain an initial guess for the trajectory. This initial guess was used for solving the final optimization problem. The results from the trajectory optimization can be seen in Figure 3. It can be seen here that an optimal time problem results in a *bang-bang* like policy for the control inputs. If an actuator is unable to realize the sharp control input changes, adding actuator constraints is trivial in the above given trajectory optimization method. For this work, we assume that the actuator can realize the given control trajectory.

5. Trajectory Stabilization

The trajectories generated using the methods given in Section 4 cannot be directly followed on a real system or in a dynamics simulator. This is due to the following: integration errors, modelling errors, disturbances, and sensor inaccuracies. To follow a trajectory, either the trajectory optimization should be run online during the maneuver, also known as Model Predictive Control (Camacho & Bordons (2007)), or an online feedback-based trajectory stabilization controller has to be utilized. In this paper, we present the latter approach and stabilize the given trajectory using a Time-Varying Linear Quadratic Regulator (TVLQR) (Bertsekas (2012)). The benefit of using TVLQR to stabilize the trajectory is that it allows for a Lyapunov stability based Region of Attraction (RoA) analysis (Reist & Tedrake (2010)), which is carried out in Section 6. For TVLQR, a time-varying linearization of the post-capture robotic system is needed. As seen in Section 4, the trajectory optimization is performed using a quaternion-based representation of the Chaser spacecraft's attitude. This choice was made as quaternions are a more compact representation when compared to rotation matrices, thereby saving memory and having fewer constraints in trajectory optimization, and are a singularity-free representation when compared to Euler angles. Free-floating multibody systems which use quaternions for rotation representation usually use either numerical differentiation (Mason et al. (2014)) or conversion to Euler angle



(a) Base Angular Velocities

(b) Joint Velocities

(c) Base Torques

(d) Joint Torques

Figure 3: Results from Detumble Trajectory Optimization.

representation (Mohamed et al. (2019)) for linearization. This is because directly linearizing quaternion-based dynamics results in a linear system that is not controllable (Zhou & Colgren (2005)). Directly linearizing the quaternion-based dynamics equations results in a linear system with 4 DoFs for the base attitude as the unit quaternion constraint is not considered. This results in an uncontrollable system. A method for quaternion-based linearization for satellite attitude control was shown in Yang (2010, 2012). In the following section, we extend this method to apply for a free-floating multibody system. This linearization is then used in the TVLQR controller synthesis.

5.1. Time Invariant Quaternion-based LQR Synthesis

A unit quaternion, used to represent rotations, can be geometrically visualized as points on a unit 3-sphere S^3 embedded in $\mathbb{R}^4$. The quaternion representing the chaser's base attitude can be then written as $\mathbf{q_b} \in \mathbb{R}^4$:

$$\mathbf{q}_b = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_4 \end{bmatrix} = \begin{bmatrix} \cos(\alpha/2) \\ \mathbf{e}\sin(\alpha/2) \end{bmatrix} \in \mathbb{R}^4 \tag{23}$$

Here, $\alpha \in [0, \pi]$ represents the equivalent rotation about about a fixed unit axis $\mathbf{e}$ (called the Euler Axis) that runs through a fixed point. This equivalency is given using the Euler's rotation theorem. By this construction, the quaternion unit length constraint $||\mathbf{q}|| = 1$ can be observed. The quaternion kinematics that describe the relation between angular velocity and quaternion derivatives can be written as shown in Equation 21c (Andrle & Crassidis, 2013).

$$\dot{\mathbf{q}}_b = \frac{1}{2}\Omega(\omega)\mathbf{q_b} = \frac{1}{2}\begin{bmatrix} 0 & -\omega^T \\ \omega & -[\omega] \end{bmatrix}\mathbf{q_b} \tag{24}$$

Here, $[\omega] \in \mathfrak{so}(3)$ is its 3×3 matrix representation (Lynch & Park, 2017). The quaternion kinematics from Equation 24 can be further expanded to it's full matrix form as:

$$\begin{bmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{25}$$

This can be rearranged as:

$$\begin{bmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}\begin{bmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{26}$$

Assuming $\alpha \neq \pm\pi$, the scalar part of the quaternion can be written as:

$$q_0 = \sqrt{1 - q_1^2 - q_2^2 - q_3^2} \tag{27}$$

We can now use this relationship in Equation 26 to get a one-to-one mapping between the angular velocities and the vector part of the quaternion derivatives:

$$\dot{\bar{\mathbf{q}}}_\mathbf{b} = \frac{1}{2}\mathbf{D_q}\omega \tag{28}$$

where $\bar{\mathbf{q}}_\mathbf{b}$ is the vector part of the unit quaternion and

$$\mathbf{D_q} = \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \tag{29}$$

It is shown in Yang (2010) that this mapping has a singularity at $\alpha = \pm\pi$. The assumption of $\alpha \neq \pm\pi$ is justified as this mapping is only used for the linearized system which is valid only in the local vicinity of the trajectory with a moving reference frame in error coordinates (as will be shown in Sub-Section 5.2). Thus, such large rotations ($\pm\pi$) are not expected in this moving relative error coordinate local reference frame along the trajectory and can be safely ignored.

A free-floating multibody system has a configuration space of $SE(3) \times \mathbb{R}^n$ where n is the number of joints in the system. The state vector for a multi-body system is written as:

$$\mathbf{x} = [\bar{\mathbf{q}}_\mathbf{b}, \mathbf{r_b}, \vartheta, \omega_\mathbf{b}, \mathbf{v_b}, \dot{\vartheta}]^T \tag{30}$$

Here, $\bar{\mathbf{q}}_b$ is the vector part of the unit quaternion representing the chaser spacecraft's floating base orientation, $\mathbf{r_b}$ is the position vector of the center of mass of the chaser spacecraft's base, ϑ is the vector of generalized joint positions, ω_b the angular velocity of the chaser spacecraft's base, $\mathbf{v}_b$ the linear velocity of the chaser spacecraft's base, and $\dot{\vartheta}$ is the generalized joint velocity vector, all expressed with respect to the inertial frame of reference, also known as hybrid representation in kinematics (Müller, 2018b). The derivative of the state can be written as:

$$\dot{\mathbf{x}} = [\dot{\bar{\mathbf{q}}}_{\mathbf{b}}, \mathbf{v}_{\mathbf{b}}, \dot{\vartheta}, \dot{\omega}_{\mathbf{b}}, \dot{\mathbf{v}}_{\mathbf{b}}, \ddot{\vartheta}]^T \tag{31}$$

From Equation 6, using a fixed point for linearization (where $\mathbf{s} = \left[\dot{\bar{\mathbf{q}}}_b, \mathbf{v}_{\mathbf{b}}, \dot{\vartheta}\right] = \mathbf{0}$), the equations of motion at the fixed point reduces to:

$$\mathbf{M}(\vartheta)\,\dot{\mathbf{s}} = \mathbf{Q}^{\text{act}} \tag{32}$$

$$\dot{\mathbf{s}} = \begin{bmatrix} \dot{\omega}_{\mathbf{b}} \\ \dot{\mathbf{v}}_{\mathbf{b}} \\ \ddot{\vartheta} \end{bmatrix} = \mathbf{M}^{-1}\mathbf{Q}^{\text{act}} \tag{33}$$

From Equation 28, Equation 31, and Equation 33 the state derivative can be written as:

$$\dot{\mathbf{x}} = [\frac{1}{2}\mathbf{D}_{\mathbf{q}}\omega_{\mathbf{b}}, \mathbf{v}_{\mathbf{b}}, \dot{\vartheta}, \mathbf{M}^{-1}\mathbf{Q}^{\text{act}}]^T \tag{34}$$

We can now take a first-order Taylor expansion of Equation 34 about a fixed point at origin

$$\begin{aligned} \dot{\mathbf{x}} \approx \mathbf{f}(\mathbf{x}^*, \mathbf{Q}^{\text{act}^*}) &+ \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*, \mathbf{Q}^{\text{act}}=\mathbf{Q}^{\text{act}^*}} (\mathbf{x} - \mathbf{x}^*) \\ &+ \left[\frac{\partial \mathbf{f}}{\partial \mathbf{Q}^{\text{act}}}\right]_{\mathbf{x}=\mathbf{x}^*, \mathbf{Q}^{\text{act}}=\mathbf{Q}^{\text{act}^*}} (\mathbf{Q}^{\text{act}} - \mathbf{Q}^{\text{act}^*}) \end{aligned} \tag{35}$$

The partial derivatives in matrix form can found in Equation A.1 and Equation A.2 in Appendix A.

After evaluating the partial derivatives at the fixed point, the linear system can be written as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{Q}^{\text{act}} \tag{36}$$

where:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n} & \frac{1}{2}\mathbf{E}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n} & \mathbf{0}_{3\times3} & \mathbf{E}_{3\times3} & \mathbf{0}_{3\times n} \\ \mathbf{0}_{n\times3} & \mathbf{0}_{n\times3} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times3} & 0_{n\times3} & \mathbf{E}_{n\times n} \\ \mathbf{0}_{(6+n)\times3} & \mathbf{0}_{(6+n)\times3} & \mathbf{0}_{(6+n)\times n} & \mathbf{0}_{(6+n)\times3} & \mathbf{0}_{(6+n)\times3} & \mathbf{0}_{(6+n)\times n} \end{bmatrix} \tag{37}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3\times(6+n)} \\ \mathbf{0}_{3\times(6+n)} \\ \mathbf{0}_{n\times(6+n)} \\ \mathbf{M}^{-1}_{(6+n)\times(6+n)} \end{bmatrix} \tag{38}$$

It can be easily verified that the controllability matrix $\begin{bmatrix}\mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \dots & \mathbf{A}^{2(6+n)-1}\mathbf{B}\end{bmatrix}$ of the linear system given above is full-rank i.e. the system is controllable. For such a controllable system LQR provides a controller which is locally optimal and globally asymptotically stable for a fully-actuated multibody system in $SE(3) \times \mathbb{R}^n$. This time invariant quaternion-based LQR is demonstrated in simulation for the system described in Section 4 by perturbing the system at a fixed point and letting the controller bring it back to origin. The initial state vector for this simulation is: $\mathbf{x}_0 = [0,0,0,0,0,0,0,0,1,0.5,1,0.1,1,0,0,0,0,1]$. The time evolution of the system can be seen in Figure 4 along with the position plots in Figure 5.

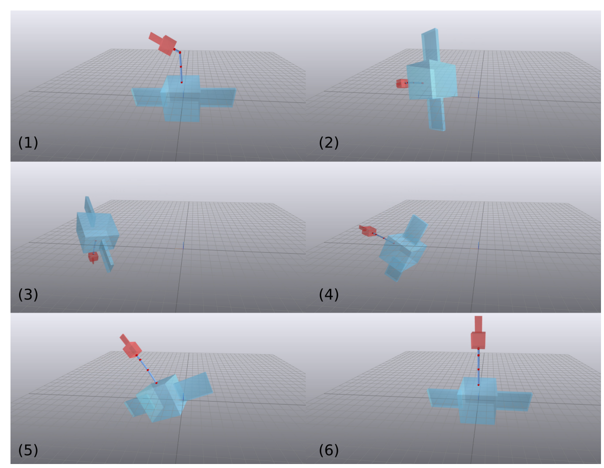
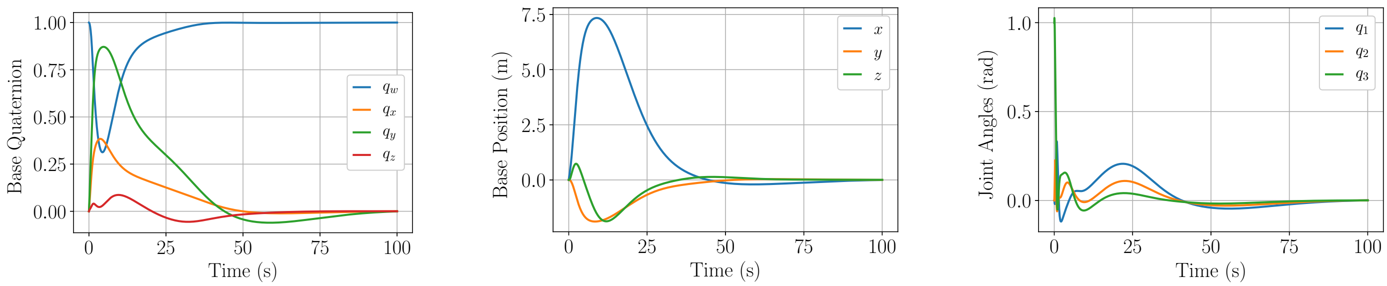


Figure 4: Time Evolution of Quaternion-Based LQR Controller stabilizing the post-capture Chaser-Target system when perturbed. The blue spacecraft with the robot arm is the chaser while the red spacecraft is the captured target. The initial perturbed state can be seen on the top left image and the stabilization follows from left to right, top to bottom with the bottom right image showing the system back to its initial fixed point.

Remark. The quaternion based linearization presented above can be alternatively performed using $SO(3)$ rotation parameterization in terms of canonical coordinates (axis/angle), and its differential. Let $\mathbf{e}$ denote the unit rotation axis vector and α denote the rotation angle. Together, $\mathbf{y} = \alpha\mathbf{e} \in \mathbb{R}^3$ defines the canonical coordinates of the first kind. In particular, the dexp mapping (Müller, 2021) $\mathbf{dexp}_{\mathbf{y}} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ provides a relationship between the angular velocity $\omega \in \mathbb{R}^3$ and the time derivative of the canonical coordinates $\dot{\mathbf{y}} \in \mathbb{R}^3$ given by:

$$\omega = \mathbf{dexp}_{\mathbf{y}}\dot{\mathbf{y}} \tag{39}$$



(a) Base Quaternions (b) Base Positions (c) Joint Positions

Figure 5: Base and Joint Positions During Stabilization using Quaternion-based LQR.

$$\text{or} \quad \dot{\mathbf{y}} = \mathbf{dexp}_{\mathbf{y}}^{-1}\omega \tag{40}$$

The closed form expression for $\mathbf{dexp}_{\mathbf{y}}^{-1}$ is given by:

$$\mathbf{dexp}_{\mathbf{y}}^{-1} = \mathbf{I} - \frac{1}{2}[\mathbf{y}] + \hat{\omega}\hat{\omega}^T(1-\gamma) \tag{41}$$

where $\gamma = \frac{\alpha'}{\beta'}, \alpha' = \operatorname{sinc}\alpha, \beta' = \operatorname{sinc}^2(\frac{\alpha}{2})$ and sinc is the cardinal sin function. Note that Equation 40 is analogous to Equation 28 introduced previously in case of quaternion based parameterization of the rotation matrix. The state vector of the free floating multi-body system with this alternative canonical parameterization is written as:

$$\mathbf{x} = [\mathbf{y}, \mathbf{r_b}, \vartheta, \omega_\mathbf{b}, \mathbf{v_b}, \dot{\vartheta}]^T \tag{42}$$

and its time derivative $\dot{\mathbf{x}}$ as

$$\dot{\mathbf{x}} = [\dot{\mathbf{y}}, \mathbf{v_b}, \dot{\vartheta}, \dot{\omega}_b, \dot{\mathbf{v}}_\mathbf{b}, \ddot{\vartheta}]^T = [\mathbf{dexp}_{\mathbf{y}}^{-1}\omega, \mathbf{v_b}, \dot{\vartheta}, \mathbf{M}^{-1}\mathbf{Q}^{\text{act}}]^T. \tag{43}$$

The corresponding $\mathbf{A}$ matrix in this case is given by:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n} & \mathbf{E}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n} & \mathbf{0}_{3\times3} & \mathbf{E}_{3\times3} & \mathbf{0}_{3\times n} \\ \mathbf{0}_{n\times3} & \mathbf{0}_{n\times3} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times3} & 0_{n\times3} & \mathbf{E}_{n\times n} \\ \mathbf{0}_{(6+n)\times3} & \mathbf{0}_{(6+n)\times3} & \mathbf{0}_{(6+n)\times n} & \mathbf{0}_{(6+n)\times3} & \mathbf{0}_{(6+n)\times3} & \mathbf{0}_{(6+n)\times n} \end{bmatrix} \tag{44}$$

and the $\mathbf{B}$ matrix remains the same as Equation 38.

5.2. Time-Varying LQR Synthesis

For creating a TVLQR controller, a time-varying error coordinate form with respect to the trajectory computed in Section 4 is defined as:

$$\bar{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}^*(t),\ \bar{\mathbf{Q}}^{\text{act}}(t) = \mathbf{Q}^{\text{act}}(t) - \mathbf{Q}^{\text{act}^*}(t) \tag{45}$$

where $\mathbf{x}^*(t)$ and $\mathbf{Q}^{\text{act}^*}(t)$ are the optimal/nominal state and control trajectories from trajectory optimization. Using the linearization methods developed in Sub-Section 5.1, we can now linearize the system along the trajectory knot points (from Section 4) using the error coordinates. However, since the system is no longer at a fixed point, we also consider the Coriolis terms in the equations of motion from Equation 6 and can re-write Equation 33 also in error coordinates as:

$$\dot{\bar{\mathbf{s}}} = \begin{bmatrix} \dot{\bar{\omega}}_\mathbf{b} \\ \dot{\bar{\mathbf{v}}}_\mathbf{b} \\ \ddot{\bar{\vartheta}} \end{bmatrix} = \mathbf{M}^{-1}(\bar{\mathbf{Q}}^{\text{act}} - \mathbf{C}\bar{\mathbf{s}}) \tag{46}$$

Here, $\bar{\mathbf{s}}$ represents the velocities in the time-varying error coordinate frame similar to in Equation 45. The state derivative from Equation 34 can also now be written in error coordinates as:

$$\dot{\bar{\mathbf{x}}} = [\frac{1}{2}\mathbf{D_q}\bar{\omega}_\mathbf{b}, \bar{\mathbf{v}}_\mathbf{b}, \dot{\bar{\vartheta}}, \mathbf{M}^{-1}(\bar{\mathbf{Q}}^{\text{act}} - \mathbf{C}\bar{\mathbf{s}})]^T \tag{47}$$

We can now take a first-order Taylor expansion of Equation 47:

$$\begin{aligned} \dot{\bar{\mathbf{x}}} = \mathbf{x}(t) - \mathbf{x}^*(t) \approx \mathbf{f}(\mathbf{x}^*, \mathbf{Q}^{\text{act}^*}) + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*, \mathbf{Q}^{\text{act}}=\mathbf{Q}^{\text{act}^*}} (\mathbf{x} - \mathbf{x}^*) \\ + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{Q}^{\text{act}}}\right]_{\mathbf{x}=\mathbf{x}^*, \mathbf{Q}^{\text{act}}=\mathbf{Q}^{\text{act}^*}} (\mathbf{Q}^{\text{act}} - \mathbf{Q}^{\text{act}^*}) - \mathbf{f}(\mathbf{x}^*, \mathbf{Q}^{\text{act}^*}) \end{aligned} \tag{48}$$

The partial derivatives in matrix form can found in Equation A.3 and Equation A.4 in Appendix A.

After evaluating the partial derivatives using automatic differentiation (Guennebaud et al., 2010), the time-varying linear system in error coordinates is obtained as:

$$\dot{\bar{\mathbf{x}}}(t) = \bar{\mathbf{A}}(t)\bar{\mathbf{x}}(t) + \bar{\mathbf{B}}(t)\bar{\mathbf{Q}}^{\text{act}}(t) \tag{49}$$

For this time-varying linear system, we use a quadratic cost to drive the errors to zero along the nominal trajectory. The quadratic cost can be divided as running cost along the trajectory and the final cost and expressed as:

$$J(\bar{\mathbf{x}},\bar{\mathbf{Q}}^{\text{act}},t) = \int_0^{t_f} (\bar{\mathbf{x}}^T(t)\mathbf{Q}\bar{\mathbf{x}}(t) + \bar{\mathbf{Q}}^{\text{act}^T}(t)\mathbf{R}\bar{\mathbf{Q}}^{\text{act}}(t))dt + \bar{\mathbf{x}}^T(t_f)\mathbf{Q_f}\bar{\mathbf{x}}(t_f) \tag{50}$$

where $\mathbf{Q} = \mathbf{Q}^T \succeq 0, \mathbf{R} = \mathbf{R}^T \succeq 0, \mathbf{Q_f} = \mathbf{Q_f}^T \succeq 0$ are positive semi-definite state and input cost, and final state cost matrices respectively. It is well known that the optimal cost-to-go J^* for such a system can be written as a time-varying quadratic term (Bertsekas, 2012; Tedrake, 2022):

$$J^*(\bar{\mathbf{x}},t) = \bar{\mathbf{x}}^T(t)\mathbf{S}(t)\bar{\mathbf{x}}(t) \tag{51}$$

Here, $\mathbf{S}(t)$ is the solution to the differential Riccati Equation. The optimal cost-to-go and $\mathbf{S}(t)$ can be obtained by solving the differential Riccati equation constructed using Equation 49 and Equation 50 backwards in time (Bertsekas, 2012; Tedrake, 2022). This is then used to construct the TVLQR gain matrix:

$$\mathbf{K}(t) = \mathbf{R}^{-1}\bar{\mathbf{B}}^T(t)\mathbf{S}(t) \tag{52}$$

The resulting controller is a time-varying optimal controller to track the given trajectory and stabilize it, this can be written as:

$$\mathbf{Q}^{\text{act}}(t) = \mathbf{Q}^{\text{act}^*}(t) - \mathbf{K}(t)\bar{\mathbf{x}}(t) \tag{53}$$

The closed loop dynamics for the simulator with the TVLQR controller can be written as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{Q}^{\text{act}^*}(t) - \mathbf{K}(t)\bar{\mathbf{x}}(t)) \tag{54}$$

The results of trajectory tracking using trajectory stabilization for detumbling can be seen in Figure 6a and Figure 6b. The actuation required by the trajectory stabilization controller is shown in Figure 6d and Figure 6e.

To examine the robustness of the TVLQR controller, the initial tumble rate of the target was varied between $4^\circ\,\mathrm{s}^{-1}$ and $6^\circ\,\mathrm{s}^{-1}$. The initial state was determined using the method provided in Sub-Section 4.2. This provides the insight into the performance of the controller with an error in the target's estimated tumble rate. The results can be seen in Figure 6c.

From Figure 6c, it can be seen that the stabilization method is robust to initial tumble rates. To further study the region of state-space that the controller can stabilize and successfully detumble, a study on the controller's RoA using Lyapunov-based probabilistic RoA estimation is given in Section 6.

6. Region of Attraction Estimation

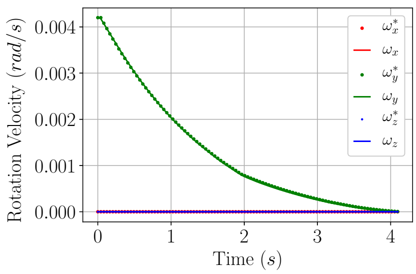
The RoA can informally be defined as the greatest area around a fixed point for which all trajectories lead towards that fixed point (Khalil, 2002). The problem of finding the RoA for nonlinear systems with a TVLQR policy can be solved by casting it as a convex sum-of-squares optimization problem (Tedrake et al., 2010) or by simulation of the nonlinear closed loop dynamics (Reist & Tedrake, 2010). Within this work, the latter method is used to obtain a probabilistic certificate that ensures the composability of sequential LQR policies. In order to estimate the RoA, the closed loop dynamics are simulated for a set of random initial conditions around the starting point of the nominal trajectory. The RoA is evaluated at a number of discrete steps k at time $k\Delta t$. The set of all estimates at these timesteps then makes up the time varying RoA which resembles a funnel.

The true RoA, albeit unknown, can conservatively be estimated by considering an invariant sublevel set of a Lyapunov function:

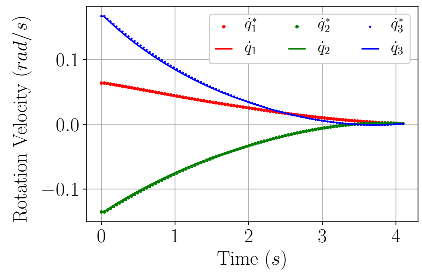
$$\mathcal{B}(\rho) = \{\mathbf{x} | V(\mathbf{x}) \leq \rho\} \tag{55}$$

Here $\mathcal{B}$ denotes a sublevel set of the Lyapunov function $V(\mathbf{x})$ that is limited by a scalar ρ. When using a TVLQR feedback policy, the optimal cost-to-go serves as a locally valid Lyapunov function Tedrake (2022). Equation 55 can then be written as:

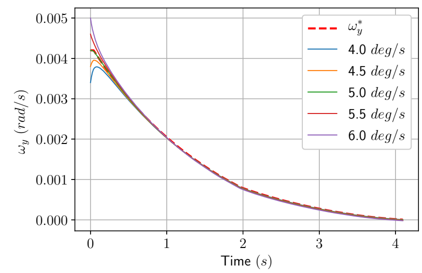
$$\mathcal{B}(\rho,t) = \{\bar{\mathbf{x}}^T(t)\mathbf{S}(t)\bar{\mathbf{x}}(t) \leq \rho(t)\} \tag{56}$$



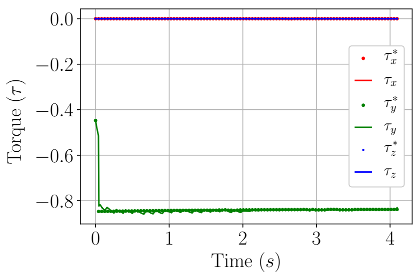
(a) Base Angular Velocities Tracking



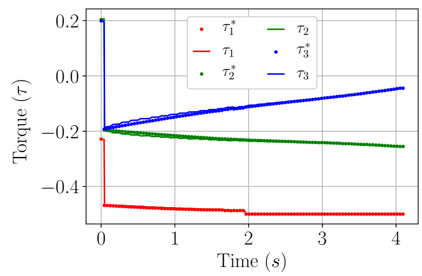
(b) Joint Velocity Tracking



(c) Detumbling of the base of the spacecraft with different target velocities.



(d) Base Actuation Torques



(e) Joint Actuation Torques

Figure 6: Results from trajectory stabilization using TVLQR to track optimal detumble trajectories. Trajectories with * represent the optimal trajectory from trajectory optimization.

Since $\mathbf{x}^*$, $\mathbf{S}$ and also ρ are functions of time, it makes sense to think about this time varying set that describes the RoA as a funnel. $\mathbf{S}(t)$ is known from solving the Ricatti Equations during TVLQR synthesis. However, in order to estimate $\mathcal{B}(\rho,t)$ an estimate of ρ is needed for every k.

Due to the free floating dynamics of the system it is sufficient for the TVLQR to bring the system into a state from which an infinite horizon Time-Invariant LQR (TILQR) policy synthesized for the fixed point at the end of the nominal trajectory could fully stabilize it. Here, we assume a policy exists that can stabilize all states within an elliptical region around the final nominal state:

$$\mathcal{B}_f = \{\mathbf{x} | \bar{\mathbf{x}}_f^T \mathbf{S}_f \bar{\mathbf{x}}_f < \rho_f\} \tag{57}$$

Here $\mathbf{S}_f = \mathbf{S}(t = t_{\text{fin}})$ is the cost-to-go matrix assigned to the last nominal state of $\mathbf{x}^*$ and $\bar{\mathbf{x}}_f = \mathbf{x} - \mathbf{x}^*(t_f)$. Furthermore, ρ_f is calculated using the maximum allowed deviation ($\bar{\mathbf{x}}_{\text{f,max}}$) from the end of the trajectory:

$$\rho_f = \bar{\mathbf{x}}_{\text{f,max}}^T \mathbf{S}_f \bar{\mathbf{x}}_{\text{f,max}} \tag{58}$$

For RoA estimation, the closed loop multibody system as defined in Equation 54 is considered. Additionally, a generalized fuel constraint limits the amount of energy that can be used for stabilization. We first simulate the nominal trajectory to obtain E_0, the nominal generalized energy, which is a time integral over the sum of all control inputs $\mathbf{Q}^{\text{act}}$. The budget for stabilization is defined with respect to E_0:

$$\frac{E_{\text{TVLQR}}}{E_0} = \alpha \tag{59}$$

Accordingly, the maximum energy $E_{\max}$ within simulations is given by the sum of the nominal energy and the contributions of the tracking controller:

$$E_{\max} = (1+\alpha)E_0 \tag{60}$$

Before the first simulation, initial conditions are drawn from a multivariate uniform distribution defined over a box shaped domain around $\mathbf{x}^*(0)$. After some failed simulations, better estimates for this region are available and only sampled states from within this updated estimate of the inlet of the funnel are considered for further processing.

The simulation is done piece-wise, from step k at time $k\Delta t$ to step $k+1$ at $(k+1)\Delta t$. If during a simulation from k to $k+1$

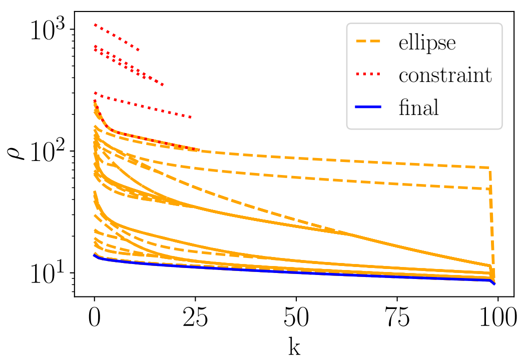


Figure 7: Evolution of ρ during the simulation runs.

a constraint is violated or, if after simulation to the subsequent slice the state is outside the last estimate of the RoA assigned to this slice, all of the preceding estimates are shrunk such that the states that lead to failure are no longer part of the RoA. If no constraint was violated and the state remained within the previously estimated RoA, the cost-to-go of this state is appended to a buffer $J^\star_{\text{buf}}$ that is used to replace RoA estimates in case a subsequent part of the current simulation fails due to the reasons previously mentioned. At step k the RoAs assigned to all preceding steps can be written as follows:

$$\rho_{j,\kappa} = \begin{cases} J^\star_{\text{buf}_\kappa}, \kappa \in \{0,\dots,k\}, & \text{if } E > E_{\max} \\ & \text{or } J^\star_{j,k+1} > \rho_{j-1,k+1} \\ \rho_{j-1,\kappa}, \kappa \in \{0,\dots,n-1\}, & \text{else} \end{cases} \tag{61}$$

This process is based on and explained in detail within Reist & Tedrake (2010).

A RoA analysis has been performed for the closed loop energy constrained system. The estimates of $\rho(t)$ over the course of 30 simulations is shown in Figure 7. During the first simulations (topmost, red lines) the fuel constraint was violated. Subsequently initial conditions with a lower initial cost-to-go were simulated (yellow), thereby continuously reducing the estimate of $\rho(t)$. The final estimate of ρ is shown by the blue line.

A more intuitive view of the RoA can be obtained by assuming that all but 2 states of x are nominal. This yields a reduced order cost-to-go formulation:

$$J^\star = \begin{bmatrix} \bar{\mathbf{x}}_p \\ \bar{\mathbf{x}}_q \end{bmatrix}^T \begin{bmatrix} \mathbf{S}_{p,p} & \mathbf{S}_{p,q} \\ \mathbf{S}_{p,q} & \mathbf{S}_{q,q} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}}_p \\ \bar{\mathbf{x}}_q \end{bmatrix} = \rho \tag{62}$$

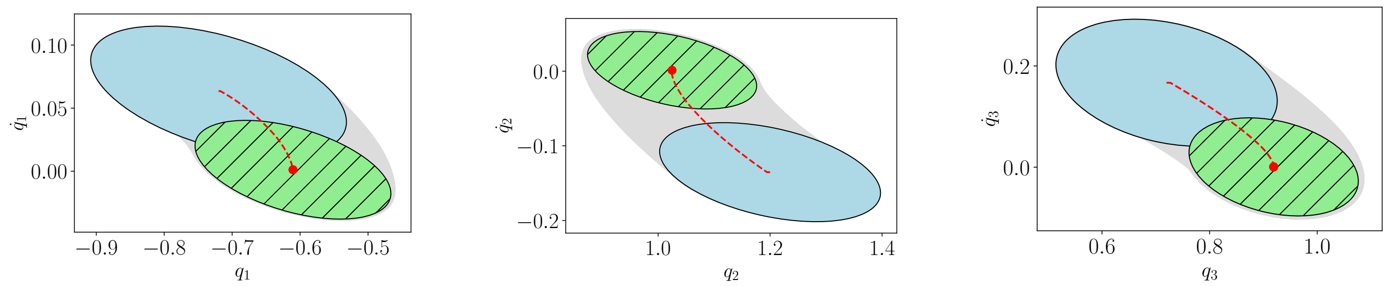
By considering an Eigendecomposition of this reduced order system for every k, a set of rotated ellipses showing a slice of the RoA around $\mathbf{x}_0(t)$ for the state variables $\mathbf{x}_p(t)$ and $\mathbf{x}_q(t)$ can be created. Figure 8a - Figure 8c depict funnels showing the RoA within various dimensions.

7. Summary

In this paper, we have introduced a method for post-capture trajectory stabilization using a Time-Varying LQR (TVLQR) controller. The initial state was computed assuming an ideal capture scenario. This initial state was then used to perform trajectory optimization to obtain an optimal detumble trajectory. The motion along the computed trajectory was stabilized using a quaternion-based TVLQR controller and tested on a dynamics simulator. The robustness of the given controller was quantified and verified using a probabilistic Region of Attraction (RoA) estimation. In contrast to other currently available methods, the RoA allows this controller to be certified for the disturbances it can recover from. This allows sequential controller composition (Burridge et al., 1999) for robotic active debris removal. It provides a goal set for the capture controller which guarantees a stable post-capture detumble. The following avenues of research will be pursued next to further this research: experimental validation using a air-bearing flat floor facility, projecting the RoA through contact dynamics to obtain the RoA in wrench space for capture, and increasing the admissible RoA using LQR-Trees.

8. Acknowledgments

This research was conducted within Stardust Reloaded project which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813644. The third author acknowledges the support of M-RoCK (Grant Number: FKZ 01IW21002) and VeryHuman (Grant Number:



(a) q_1 vs. $\dot{q}_1$ (b) q_2 vs. $\dot{q}_2$ (c) q_3 vs. $\dot{q}_3$

Figure 8: Estimated RoA within the state space. Every nominal initial state from within the blue ellipse leads to a final state with a cost-to-go that is smaller than that of the states associated with the border of the green, hatched ellipse.

FKZ 01IW20004) projects funded by the German Aerospace Center (DLR) with federal funds from the Federal Ministry of Education and Research (BMBF). The fifth author acknowledges that this work has been supported by the 'LCM-K2 Center for Symbiotic Mechatronics' within the framework of the Austrian COMET-K2 program.

Appendix A. Linearized System Matrices

$$\mathbf{A} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right] = \begin{bmatrix} \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \bar{\mathbf{q}}_\mathbf{b}} & \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \mathbf{r_b}} & \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \boldsymbol{\vartheta}} & \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \boldsymbol{\omega}_\mathbf{b}} & \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \mathbf{v_b}} & \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \dot{\boldsymbol{\vartheta}}} \\ \frac{\partial \mathbf{v_b}}{\partial \bar{\mathbf{q}}_\mathbf{b}} & \frac{\partial \mathbf{v_b}}{\partial \mathbf{r_b}} & \frac{\partial \mathbf{v_b}}{\partial \boldsymbol{\vartheta}} & \frac{\partial \mathbf{v_b}}{\partial \boldsymbol{\omega}_\mathbf{b}} & \frac{\partial \mathbf{v_b}}{\partial \mathbf{v_b}} & \frac{\partial \mathbf{v_b}}{\partial \dot{\boldsymbol{\vartheta}}} \\ \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \bar{\mathbf{q}}_\mathbf{b}} & \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \mathbf{r_b}} & \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \boldsymbol{\vartheta}} & \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \boldsymbol{\omega}_\mathbf{b}} & \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \mathbf{v_b}} & \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \dot{\boldsymbol{\vartheta}}} \\ \frac{\partial \mathbf{M}^{-1}\mathbf{Q}^{\mathrm{act}}}{\partial \bar{\mathbf{q}}_\mathbf{b}} & \frac{\partial \mathbf{M}^{-1}\mathbf{Q}^{\mathrm{act}}}{\partial \mathbf{r_b}} & \frac{\partial \mathbf{M}^{-1}\mathbf{Q}^{\mathrm{act}}}{\partial \boldsymbol{\vartheta}} & \frac{\partial \mathbf{M}^{-1}\mathbf{Q}^{\mathrm{act}}}{\partial \boldsymbol{\omega}_\mathbf{b}} & \frac{\partial \mathbf{M}^{-1}\mathbf{Q}^{\mathrm{act}}}{\partial \mathbf{v_b}} & \frac{\partial \mathbf{M}^{-1}\mathbf{Q}^{\mathrm{act}}}{\partial \dot{\boldsymbol{\vartheta}}} \end{bmatrix} \tag{A.1}$$

$$\mathbf{B} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{Q}^{\mathrm{act}}}\right] = \begin{bmatrix} \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \mathbf{Q}^{\mathrm{act}}} \\ \frac{\partial \mathbf{v_b}}{\partial \mathbf{Q}^{\mathrm{act}}} \\ \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \mathbf{Q}^{\mathrm{act}}} \\ \frac{\partial \mathbf{M}^{-1}\mathbf{Q}^{\mathrm{act}}}{\partial \mathbf{Q}^{\mathrm{act}}} \end{bmatrix} \tag{A.2}$$

$$\mathbf{A}(t) = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right] = \begin{bmatrix} \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \bar{\mathbf{q}}_\mathbf{b}} & \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \mathbf{r_b}} & \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \boldsymbol{\vartheta}} & \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \boldsymbol{\omega}_\mathbf{b}} & \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \mathbf{v_b}} & \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \dot{\boldsymbol{\vartheta}}} \\ \frac{\partial \mathbf{v_b}}{\partial \bar{\mathbf{q}}_\mathbf{b}} & \frac{\partial \mathbf{v_b}}{\partial \mathbf{r_b}} & \frac{\partial \mathbf{v_b}}{\partial \boldsymbol{\vartheta}} & \frac{\partial \mathbf{v_b}}{\partial \boldsymbol{\omega}_\mathbf{b}} & \frac{\partial \mathbf{v_b}}{\partial \mathbf{v_b}} & \frac{\partial \mathbf{v_b}}{\partial \dot{\boldsymbol{\vartheta}}} \\ \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \bar{\mathbf{q}}_\mathbf{b}} & \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \mathbf{r_b}} & \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \boldsymbol{\vartheta}} & \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \boldsymbol{\omega}_\mathbf{b}} & \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \mathbf{v_b}} & \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \dot{\boldsymbol{\vartheta}}} \\ \frac{\partial \mathbf{M}^{-1}(\bar{\mathbf{Q}}^{\mathrm{act}} - \mathbf{C}\bar{\mathbf{s}})}{\partial \bar{\mathbf{q}}_\mathbf{b}} & \frac{\partial \mathbf{M}^{-1}\mathbf{M}^{-1}(\bar{\mathbf{Q}}^{\mathrm{act}} - \mathbf{C}\bar{\mathbf{s}})}{\partial \mathbf{r_b}} & \frac{\partial \mathbf{M}^{-1}(\bar{\mathbf{Q}}^{\mathrm{act}} - \mathbf{C}\bar{\mathbf{s}})}{\partial \boldsymbol{\vartheta}} & \frac{\partial \mathbf{M}^{-1}(\bar{\mathbf{Q}}^{\mathrm{act}} - \mathbf{C}\bar{\mathbf{s}})}{\partial \boldsymbol{\omega}_\mathbf{b}} & \frac{\partial \mathbf{M}^{-1}(\bar{\mathbf{Q}}^{\mathrm{act}} - \mathbf{C}\bar{\mathbf{s}})}{\partial \mathbf{v_b}} & \frac{\partial \mathbf{M}^{-1}(\bar{\mathbf{Q}}^{\mathrm{act}} - \mathbf{C}\bar{\mathbf{s}})}{\partial \dot{\boldsymbol{\vartheta}}} \end{bmatrix} \tag{A.3}$$

$$\mathbf{B}(t) = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{Q}^{\mathrm{act}}}\right] = \begin{bmatrix} \frac{\partial \frac{1}{2}\mathbf{D_q}\boldsymbol{\omega}_\mathbf{b}}{\partial \mathbf{Q}^{\mathrm{act}}} \\ \frac{\partial \mathbf{v_b}}{\partial \mathbf{Q}^{\mathrm{act}}} \\ \frac{\partial \dot{\boldsymbol{\vartheta}}}{\partial \mathbf{Q}^{\mathrm{act}}} \\ \frac{\partial \mathbf{M}^{-1}(\bar{\mathbf{Q}}^{\mathrm{act}} - \mathbf{C}\bar{\mathbf{s}})}{\partial \mathbf{Q}^{\mathrm{act}}} \end{bmatrix} \tag{A.4}$$

References

Aghili, F. (2008). Optimal control for robotic capturing and passivation of a tumbling satellite with unknown dynamics. In *Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit* (p. 7274). Honolulu, Hawaii: AIAA.

Aghili, F. (2009a). Coordination control of a free-flying manipulator and its base attitude to capture and detumble a noncooperative satellite. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2365–2372). IEEE. doi:10.1109/IROS.2009.5353968.

Aghili, F. (2009b). Optimal control of a space manipulator for detumbling of a target satellite. In *Proceedings of IEEE International Conference on Robotics and Automation* (pp. 3019–3024). IEEE. doi:10.1109/ROBOT.2009.5152235.

Aghili, F. (2009c). Time-optimal detumbling control of spacecraft. *Journal of guidance, control, and dynamics*, *32*(5), 1671–1675.

Aghili, F. (2010). Cartesian control of space manipulators for on-orbit servicing. In *Proceedings of AIAA Guidance, Navigation, and Control Conference* (p. 7846). AIAA.

Aghili, F. (2013). Pre- and post-grasping robot motion planning to capture and stabilize a tumbling/drifting free-floater with uncertain dynamics. In *Proceedings of IEEE International Conference on Robotics and Automation* (pp. 5461–5468). doi:10.1109/ICRA.2013.6631360.

Aghili, F. (2020). Optimal trajectories and robot control for detumbling a non-cooperative satellite. *Journal of Guidance, Control, and Dynamics*, *43*(5), 981–988. doi:10.2514/1.G004758.

Andrle, M. S., & Crassidis, J. L. (2013). Geometric integration of quaternions. *Journal of Guidance, Control, and Dynamics*, *36*(6), 1762–1767. doi:10.2514/1.58558.

Anz-Meador, P. D. (2020). Orbital debris Quarterly News. *NASA*, *24*(1).

Bertsekas, D. P. (2012). *Dynamic programming and optimal control: Volume. 1*. Nashua, US: Athena scientific.

Betts, J. T. (2010). *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Philadelphia: Society for Industrial and Applied Mathematics. doi:10.1137/1.9780898718577.

Brockett, R. W. (1984). Robotic manipulators and the product of exponentials formula. In *Mathematical Theory of Networks and Systems* (pp. 120–129). Springer Berlin Heidelberg.

Burridge, R. R., Rizzi, A. A., & Koditschek, D. E. (1999). Sequential composition of dynamically dexterous robot behaviors. *International Journal of Robotics Research*, *18*(6), 534–555. doi:10.1177/02783649922066385.

Camacho, E. F., & Bordons, C. (2007). *Model Predictive control*. Advanced Textbooks in Control and Signal Processing. London: Springer London. doi:10.1007/978-0-85729-398-5.

Dimitrov, D. N., & Yoshida, K. (2004a). Momentum distribution in a space manipulator for facilitating the post-impact control. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3345–3350). IEEE. doi:10.1109/iros.2004.1389933.

Dimitrov, D. N., & Yoshida, K. (2004b). Utilization of the bias momentum approach for capturing a tumbling satellite. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3333–3338). IEEE. doi:10.1109/iros.2004.1389931.

Dubowsky, S., & Papadopoulos, E. (1993). The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *IEEE Transactions on robotics and automation*, *9*(5), 531–543. doi:10.1109/70.258046.

ESA Space Debris Office (2022). *ESA's Annual Space Environment Report*. Technical Report April ESA. URL: https://bit.ly/2Soiqfi.

Featherstone, R. (2008). *Rigid Body Dynamics Algorithms*. Boston, MA: Springer US. doi:https://doi.org/10.1007/978-1-4899-7560-7.

Flores-Abad, A., Zhang, L., Wei, Z. et al. (2017). Optimal capture of a tumbling object in orbit using a space manipulator. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *86*(2), 199–211. doi:10.1007/s10846-016-0417-1.

Gill, P. E., Murray, W., & Saunders, M. A. (2005). SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, *47*(1), 99–131.

Gómez, N. O., & Walker, S. J. (2015). Eddy currents applied to de-tumbling of space debris: Analysis and validation of approximate proposed methods. *Acta Astronautica*, *114*, 34–53.

Graham, J., Ravindran, R., & Knapp, K. (1979). Space manipulators - Present capability and future potential. In *Proceedings of Conference on Advanced Technology for Future Space Systems* (pp. 243–253). Reston, Virigina: AIAA. doi:10.2514/6.1979-903.

Guennebaud, G., Jacob, B., Munk Larsen, R. et al. (2010). Eigen v3. http://eigen.tuxfamily.org.

Hildebrandt, M., Albiez, J., & Kirchner, F. (2008). Computer-based control of deep-sea manipulators. In *Proceedings of OCEANS MTS/IEEE Kobe Techno-Ocean* (pp. 1–6). IEEE. doi:10.1109/OCEANSKOBE.2008.4531026.

Jain, A. (2011). *Robot and Multibody Dynamics: Analysis and Algorithms*. Boston, MA: Springer US.

Jankovic, M., & Kirchner, F. (2018). Trajectory generation method for robotic free-floating capture of a non-cooperative, tumbling target. In *Proceedings of Stardust Final Conference* (pp. 111–127). Springer.

Jankovic, M., Yüksel, M., Babr, M. M. et al. (2020). Space debris ontology for ADR capture methods selection. *Acta Astronautica*, *173*, 56–68.

Kelly, M. (2017). An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, *59*(4), 849–904. URL: https://github.com/MatthewPeterKelly/OptimTraj. doi:10.1137/16M1062569.

Khalil, H. K. (2002). *Nonlinear Systems*. (3rd ed.). Upper Saddle River, N.J: Prentice Hall.

Liou, J. C. (2011). An active debris removal parametric study for LEO environment remediation. *Advances in Space Research*, *47*(11), 1865–1876. doi:10.1016/j.asr.2011.02.003.

Lynch, K. M., & Park, F. (2017). *Modern Robotics: Mechanics, Planning, and Control*. Cambridge: Cambridge University Press.

Mark, C. P., & Kamath, S. (2019). Review of active space debris removal methods. *Space Policy*, *47*, 194–206. doi:10.1016/j.spacepol.2018.12.005.

Mason, S., Righetti, L., & Schaal, S. (2014). Full dynamics LQR control of a humanoid robot: An experimental study on balancing and squatting. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots* (pp. 374–379). IEEE. doi:10.1109/HUMANOIDS.2014.7041387.

Matunaga, S., Kanzawa, T., & Ohkami, Y. (2001). Rotational motion-damper for the capture of an uncontrolled floating satellite. *Control Engineering Practice*, *9*(2), 199–205. doi:10.1016/S0967-0661(00)00098-8.

Mohamed, A., Saaj, C., Seddaoui, A. et al. (2019). Controlling a non-linear space robot using linear controllers. In *Proceedings of 5th CEAS Conference on Guidance, Navigation and Control (EuroGNC)*. Milan, Italy: CEAS.

Müller, A. (2018a). Screw and lie group theory in multibody dynamics. *Multibody System Dynamics*, *42*(2), 219–248.

Müller, A. (2018b). Screw and lie group theory in multibody kinematics. *Multibody System Dynamics*, *43*(1), 37–70.

Müller, A. (2021). Review of the exponential and cayley map on se(3) as relevant for lie group integration of the generalized poisson equation and flexible multibody systems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *477*(2253), 20210303. doi:10.1098/rspa.2021.0303.

Nanos, K., & Papadopoulos, E. (2012). On cartesian motions with singularities avoidance for free-floating space robots. In *Proceedings of IEEE International Conference on Robotics and Automation* (pp. 5398–5403). IEEE.

Nenchev, D. N., & Yoshida, K. (1999). Impact analysis and post-impact motion control issues of a free-floating space robot subject to a force impulse. *IEEE Transactions on Robotics and Automation*, *15*(3), 548–557. doi:10.1109/70.768186.

Orin, D. E., & Goswami, A. (2008). Centroidal momentum matrix of a humanoid robot: Structure and properties. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 653–659). IEEE. doi:10.1109/IROS.2008.4650772.

Orin, D. E., Goswami, A., & Lee, S.-H. (2013). Centroidal dynamics of a humanoid robot. *Autonomous robots*, *35*(2), 161–176.

Papadopoulos, E., & Dubowsky, S. (1991). On the nature of control algorithms for free-floating space manipulators. *IEEE Transactions on Robotics and Automation*, *7*(6), 750–758. doi:10.1109/70.105384.

Papadopoulos, E., & Dubowsky, S. (1993a). Dynamic Singularities in Free-floating Space Manipulators. In Y. Xu, & T. Kanade (Eds.), *Space Robotics: Dynamics and Control* (pp. 77–100). Boston, MA: Springer US volume 188. doi:10.1007/978-1-4615-3588-1_4.

Papadopoulos, E., & Dubowsky, S. (1993b). Dynamic singularities in free-floating space manipulators. *Journal of dynamic systems, measurement, and control*, *115*(1), 44–52.

Park, F., Bobrow, J., & Ploen, S. (1995). A lie group formulation of robot dynamics. *The International Journal of Robotics Research*, *14*(6), 609–618. doi:10.1177/027836499501400606.

Raina, D., Gora, S., Maheshwari, D. et al. (2021). Impact modeling and reactionless control for post-capturing and maneuvering of orbiting objects using a multi-arm space robot. *Acta Astronautica*, *182*, 21–36. doi:10.1016/j.actaastro.2021.01.034.

Reist, P., & Tedrake, R. (2010). Simulation-based LQR-trees with input and state constraints. In *Proceedings of IEEE International Conference on Robotics and Automation* (pp. 5504–5510). IEEE. doi:10.1109/ROBOT.2010.5509893.

Rybus, T., Seweryn, K., & Sąsiadek, J. Z. (2014). Optimal detumbling of defunct spacecraft using space robots. In *Proceedings of 19th International Conference on Methods and Models in Automation and Robotics, MMAR 2014* (pp. 64–69). IEEE. doi:10.1109/MMAR.2014.6957326.

Rybus, T., Seweryn, K., & Sasiadek, J. Z. (2016). Trajectory Optimization of Space Manipulator with Non-zero Angular Momentum During Orbital Capture Maneuver. In *Proceedings of AIAA Guidance, Navigation, and Control Conference*. AIAA. doi:10.2514/6.2016-0885.

Shibli, M., Aghili, F., & Su, C.-Y. (2006). Hybrid inverse dynamics control of a free-flying space robot in contact with a target satellite. In *Proceedings of 1st International Symposium on Systems and Control in Aerospace and Astronautics* (pp. 6 pp.–6). IEEE. doi:10.1109/ISSCAA.2006.1627691.

Tedrake, R. (2022). *Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation*. URL: http://underactuated.mit.edu.

Tedrake, R., & the Drake Development Team (2019). Drake: Model-based design and verification for robotics. URL: https://drake.mit.edu.

Tedrake, R., Manchester, I. R., Tobenkin, M. et al. (2010). LQR-trees: Feedback Motion Planning via Sums-of-Squares Verification. *The International Journal of Robotics Research*, *29*(8), 1038–1052. doi:10.1177/0278364910369189.

Umetani, Y., & Yoshida, K. (1987). Continuous path control of space manipulators mounted on omv. *Acta Astronautica*, *15*(12), 981–986.

Umetani, Y., & Yoshida, K. (1989). Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix. *IEEE Transactions on Robotics and Automation*, *5*(3), 303–314. doi:10.1109/70.34766.

Vafa, Z., & Dubowsky, S. (1990). The Kinematics and Dynamics of Space Manipulators: The Virtual Manipulator Approach. *The International Journal of Robotics Research*, *9*(4), 3–21. doi:10.1177/027836499000900401.

Vetrisano, M., Thiry, N., & Vasile, M. (2015). Detumbling large space debris via laser ablation. In *Proceedings of IEEE Aerospace Conference* (pp. 1–10). IEEE. doi:10.1109/AERO.2015.7119051.

Wang, M., Luo, J., Yuan, J. et al. (2018a). Detumbling strategy and coordination control of kinematically redundant space robot after capturing a tumbling target. *Nonlinear Dynamics*, *92*(3), 1023–1043. doi:10.1007/s11071-018-4106-4.

Wang, M. M., Luo, J. J., Yu, M. et al. (2018b). Detumbling Control for Kinematically Redundant Space Manipulator Post-Grasping a Rotational Satellite. *Journal of Astronautics*, *39*(5), 550–561. doi:10.3873/j.issn.1000-1328.2018.05.010.

Xu, Y. (1993). The measure of dynamic coupling of space robot systems. In *Proceedings of IEEE International Conference on Robotics and Automation* (pp. 615–620). IEEE. doi:10.1109/ROBOT.1993.291837.

Yang, Y. (2010). Quaternion based model for momentum biased nadir pointing spacecraft. *Aerospace Science and Technology*, *14*(3), 199–202. doi:10.1016/j.ast.2009.12.006.

Yang, Y. (2012). Analytic LQR Design for Spacecraft Control System Based on Quaternion Model. *Journal of Aerospace Engineering*, *25*(3), 448–453. doi:10.1061/(asce)as.1943-5525.0000142.

Yoshida, K., & Nenchev, D. N. (1998). A general formulation of under-actuated manipulator systems. In *Proceedings of Robotics Research* (pp. 33–44). Springer London. doi:10.1007/978-1-4471-1580-9_4.

Yoshida, K., & Umetani, Y. (1993). Control of Space Manipulators with Generalized Jacobian Matrix. In *Space Robotics: Dynamics and Control* (pp. 165–204). Boston, MA: Springer US. doi:10.1007/978-1-4615-3588-1_7.

Zhang, B., Liang, B., Wang, Z. et al. (2017). Coordinated stabilization for space robot after capturing a noncooperative target with large inertia. *Acta Astronautica*, *134*, 75–84. doi:10.1016/j.actaastro.2017.01.041.

Zhou, Z., & Colgren, R. (2005). A non-linear spacecraft attitude tracking controller for large non-constant rate commands. *International Journal of Control*, *78*(5), 311–325. doi:10.1080/00207170500079779.