Quaternion based LQR for Free-Floating Robots Without Gravity

Shubham Vyas Researcher, Robotics Innovation Center, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, 28359, Bremen, Germany. Shubham.Vyas@dfki.de

Bilal Wehbe Researcher, Robotics Innovation Center, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, 28359, Bremen, Germany. Bilal.Wehbe@dfki.de

Shivesh Kumar Researcher, Robotics Innovation Center, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, 28359, Bremen, Germany. Shivesh.Kumar@dfki.de

ABSTRACT

Quaternions are commonly used for rotation representation as they avoid the singularities found in the Euler angles representation and are more compact than using rotation matrices (for storage, operations, and constraints required). However, it is difficult to use quaternions in linear control approaches due to the inherent unit length constraint of the representation. Quaternion-based linear control has been previously used for single rigid body control such as quadrotors and satellite attitude control. In this paper, we provide an analytical method for linearizing multibody free-floating robotic systems without gravity using a quaternion-based rotation representation for the floating base. This linearization is then used for deriving a Linear Quadratic Regulator (LQR) based controller. The LQR is optimal in the local neighbourhood of the linearization and is globally asymptotically stable for such systems. The utility of this method is demonstrated using two examples from different robotic domains: space and underwater robotics.

Keywords: Robotics;Space Robotics; Underwater Robotics; LQR; Linear Optimal Control; Quaternions

1 Introduction

Quaternions are a popular way of rotation representation for robots with free-floating base such as quadrotors, planes, space robotic systems, underwater systems, and legged robots. This is due to their advantages over Euler angles (singularity/gimbal-lock free and order independent representation) and over rotation matrices (more compact representation, fewer floating point operations during rotational transformations and increased numerical stability). Furthermore, they provide a natural distance metric for rotations. Due to these reasons, quaternions have found widespread use in trajectory optimization [1, 2] and control [3–9]. Due to the advantages offered by quaternions, Quaternion-based control has been investigated for a long time. Some of the investigations into non-linear control using quaternions for stability analysis using Lyapunov include [8, 9]. Finding Lyapunov functions for the quaternion-based non-linear systems is challenging and generally driven by intuition [10]. Furthermore, Lyapunov functions provide guarantees on the stability of the system but they do not deliberate about the optimality of the system. Optimal control of the robotic systems is a practical goal for many applications. For non-linear systems, a way to carry out both stability and optimality analysis is by linearizing them about the required operating point. Linear systems and their analysis has a rich history of both optimality and

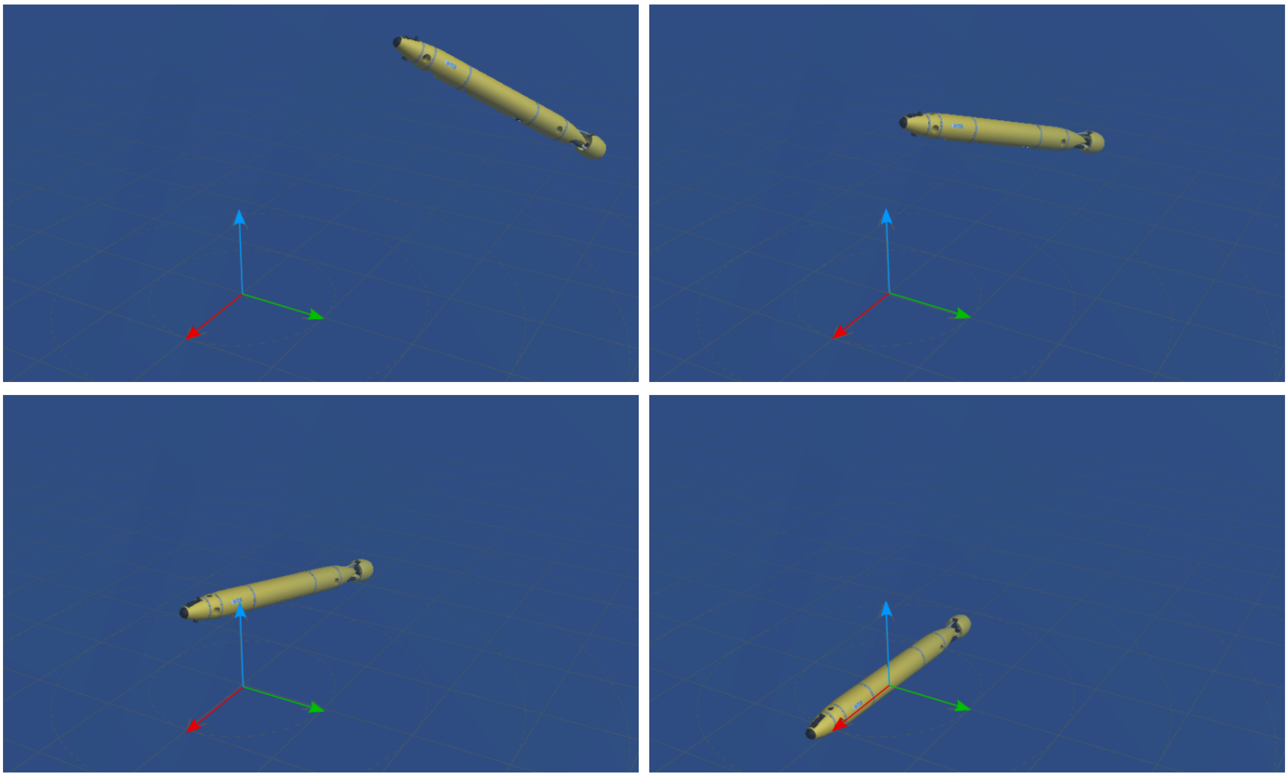


Figure 1 Figure showing different snapshot of the AUV during a control maneuver. The top-left figure shows the initial position, top-right and bottom-left figures show the transition, and the bottom-right image shows the vehicle's final state.

stability. Multi-body systems with a free-floating base such as a humanoids and space robots also benefit from the use of linear model and LQR-based control as shown in [7, 11]. However, until now, these either use numerical differentiation [7] or Euler representation (after conversion to Euler angles) [11] for synthesizing the LQR controller. Having a linearized model of the dynamics allows the use of rich literature of linear systems to analyze the system from multiple perspectives regarding optimality and stability. Control techniques such as pole placement, LQR, and H_∞ can be used for linear systems to drive the system to the desired state in an optimal manner. However, directly linearizing quaternion-based dynamics results in a linear system that is not controllable [12]. This is due to the quaternion unit length constraint which the linear system has no knowledge about. It therefore tries to control the components individually while they are, in actuality, constrained. This prevents the use of linear system analysis and control tools while using quaternions. A method for successfully linearizing the quaternion dynamics for the attitude control of a rigid body in $\mathbb{SO}(3)$ was presented in [3, 13]. It embeds the unit quaternion constraint to create a reduced model of the quaternion-based dynamics which is controllable. Similar work can be seen in the quadrotor controls community [4–6, 14, 15]. These results are demonstrated for the attitude control of a single rigid body in $\mathbb{SO}(3)$.

In this paper, we summarize and derive the linearization as given in the literature for control of a single rigid body in $\mathbb{SO}(3)$ and $\mathbb{SE}(3)$ using an LQR. We further extend this method to provide an analytical method to synthesize an quaternion-based LQR controller for multi-body systems without gravity. The application use case presented for this is the problem of post-capture detumbling in a robotic space debris removal mission. Furthermore, we improve the fidelity of the single rigid body linearization for control of autonomous underwater vehicles (AUV) by including additional hydrodynamic bias terms to the linearization and demonstrate control of the AUV (see Figure 1 for an example). The methods presented here allows the controller synthesis to be free from numerical errors and representation transformations. Furthermore, combining the given methods with quaternion-based trajectory optimization methods from

[1, 2] can yield a control pipeline using quaternions independent from other rotation representations which could induce errors/confusion.

The layout of this paper is as follows, Section 2 describes the required background mathematical preliminaries, Section 3 provides the derivation for analytical quaternion based linearization used for LQR along with additional bias terms. Section 4 demonstrates the applications of the derived linear model in an LQR controller for space and underwater application scenarios, and finally Section 5 presents the conclusion and future work.

2 Rotational Representations

Any rotation is a motion of a certain space that preserves at least one point. All rotations about a fixed point form a group. In rigid body kinematics, they form a matrix Lie group known as the special orthogonal group $\mathbb{SO}(n)$

$$\mathbb{SO}(n) = \{R \in \mathbb{GL}(n,\mathbb{F}) : RR^T = I, \det R = +1\}$$

where n denotes the dimension of the group, $\mathbb{GL}(n,\mathbb{F})$ is the general linear group defined on the underlying set $\mathbb{F}$ with matrix multiplication as the group operation. An element in $\mathbb{SO}(n)$ can be parameterized by $n(n-1)/2$ unique parameters. Out of n^2 parameters of a general real matrix, the remaining $n(n+1)/2$ parameters are determined by the condition $RR^T = I$. For example, the matrix representation of 3 dimensional rotation $\mathbb{SO}(3)$ requires $3 \times 3 = 9$ parameters out of which 3 are unique. The Lie algebra of $\mathbb{SO}(3)$, denoted as $\mathfrak{so}(3)$, may be identified with the 3×3 skew-symmetric matrices of the form

$$[\omega] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{1}$$

with the Lie Bracket structure

$$[[\omega_1],[\omega_2]] = [\omega_1][\omega_2] - [\omega_2][\omega_1] \tag{2}$$

where $[\omega_1],[\omega_2] \in \mathfrak{so}(3)$. We can identity $\mathfrak{so}(3)$ with $\mathbb{R}^3$ using the Eqn. (1), which maps a vector $\omega \in \mathbb{R}^3$ to a matrix $[\omega] \in \mathfrak{so}(3)$. It is straight forward to show that

$$[[\omega_1],[\omega_2]] = [\omega_1 \times \omega_2] \tag{3}$$

where $\omega_1, \omega_2 \in \mathbb{R}^3$. Thus, $\omega \mapsto [\omega]$ is a Lie algebra isomorphism between the Lie algebra $(\mathbb{R}^3, \times)$ and the Lie algebra $(\mathfrak{so}(3), [\cdot,\cdot])$. Let $R(t)$ and $\dot{R}$ denote the time dependent orientation of the rotating frame and its rate of change w.r.t. time respectively as seen from a fixed inertial frame. Denote by ω the angular velocity of the rotating frame. Then

$$\dot{R}R^{-1} = [\omega_s], \quad R^{-1}\dot{R} = [\omega_b] \tag{4}$$

where $\omega_s, \omega_b \in \mathbb{R}^3$ are the *space fixed* and *body fixed* vector representations of ω respectively, and $[\omega] \in \mathfrak{so}(3)$ is its 3×3 matrix representation [16].

Many alternative compact parameterizations of the 3 dimensional rotation are available in the literature. These include minimal parameterizations like roll-pitch-yaw angles, Euler angles, Caylay-Rodriguez parameters etc., but they are known to suffer from singularities. Unit quaternion is a compact singularity-free way to represent rotations with only 4 parameters. Euler's rotation theorem states that any rotation or sequence of rotations of a rigid body or coordinate system about a fixed point is equivalent to a single rotation by a given angle $\alpha \in [0,\pi]$ about a fixed unit axis $\hat{\omega}$ (called the Euler axis) that runs through the fixed point. The unit quaternion representation of $R = \exp([\hat{\omega}]\alpha)$ is constructed as follows. Denote $q \in \mathbb{R}^4$ according to

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_4 \end{bmatrix} = \begin{bmatrix} \cos(\alpha/2) \\ \hat{\omega}\sin(\alpha/2) \end{bmatrix} \in \mathbb{R}^4 \tag{5}$$

The unit normalization constraint $||q|| = 1$ follows from its construction. Geometrically, unit quaternions q can be visualized as points on a unit 3-sphere S^3 embedded in $\mathbb{R}^4$. Using the identity $1 + 2\cos(\alpha) = \mathrm{tr}(R)$ (with $\mathrm{tr}(R)$ representing trace of the matrix R) and the cosine double angle formula $\cos 2\phi = 2\cos^2\phi - 1$, the elements of q can be obtained directly from the entries of R as follows:

$$q_0 = \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}}, \quad \begin{bmatrix} q_0 \\ q_1 \\ q_2 \end{bmatrix} = \frac{1}{4q_0}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}.$$

On the contrary, given a unit quaternion $q = [q_0, q_1, q_2, q_3]$, the corresponding rotation matrix R is obtained as a rotation about the unit axis, in the direction of (q_1, q_2, q_3) by an angle $2\cos^{-1}(q_0)$ as follows:

$$\begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 - q_1q_3) \\ 2(q_0q_3 + q_1q_2) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_1q_0) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}.$$

It is apparent from the above formula that both $q \in S^3$ and its antipodal point $-q \in S^3$ produce the same rotation matrix R. This is known as the double cover property of quaternions. Additionally, quaternions are isomorphic to special unitary group $\mathbb{SU}(2)$. The mapping between angular velocity and the quaternion derivatives i.e. quaternion kinematics is given by the following relation [17]:

$$\dot{q}_b = \frac{1}{2}\Omega(\omega)q_b \tag{6}$$

This is a differential equation on $\mathbb{SU}(2)$. Here, the matrix operator $\frac{1}{2}\Omega(\omega)$ is given as:

$$\frac{1}{2}\Omega(\omega) = \frac{1}{2}\begin{bmatrix} 0 & -\omega^T \\ \omega & -[\omega] \end{bmatrix} \tag{7}$$

Quaternions have become increasingly popular in various application areas for e.g. computer graphics, computer vision, robotics, navigation, flight dynamics, orbital mechanics of satellites etc. The main reasons of this popularity are its compact singularity free representation and computational efficient quaternion algebra in comparison to working with rotation matrices. Some additional advantages include spherical interpolation and the existence of a simple distance metric.

3 Quaternion-Based Linearization

In this section, we extend the method given in [3]. The method given in [3] provides an LQR controller for a single rigid body in $\mathbb{SO}(3)$. We first extend this to a single rigid body in $\mathbb{SE}(3)$ and then further extend it to a multi-body system made of rigid links in $\mathbb{SE}(3) \times \mathbb{R}^n$. The equations given in this section are, in general, frame invariant as long as all the terms are computed with respect to the same reference frame. We describe the frames in which they are described in for easier implementation of the example applications in Section 4.

3.1 Mapping Angular Velocities to Quaternion Derivatives

Let us consider a unit quaternion q_b, which represents a rotation α about the given unitary rotation axis $\hat{\omega}$. The quaternion kinematic equation from Equation 6 can be expanded to its full matrix form as:

$$\begin{bmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{8}$$

This equation can be rearranged as:

$$\begin{bmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{9}$$

For a given unit-quaternion representation of a rotation q_b, if the rotation $\alpha \neq \pm\pi$, the scalar part q_0 can be written as:

$$q_0 = \sqrt{1 - q_1^2 - q_2^2 - q_3^2} \tag{10}$$

We can substitute this in Equation 9 to get a one-to-one mapping between the angular velocities and quaternion derivatives (only the vector part) which can be written as:

$$\dot{q}_b = \frac{1}{2} Q\omega \tag{11}$$

where:

$$Q = \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \tag{12}$$

As shown in [13], this one-to-one mapping has a singularity at $\alpha = \pm\pi$. However, we use this mapping for the linearized version of the dynamic system. A linearized version of the non-linear system is only used for control in the local neighbourhood of the linearization point where the linearization assumptions are valid. We do not expect such large rotations to be seen during the use of the linear system and thus the singularities at $\pm\pi$ can safely be ignored for the purpose of this paper. This mapping is used to perform the linearization and synthesize an optimal linear controller.

3.2 Single Rigid Body Linearization using Quaternions

By utilizing Equation 10, we can write the full state of a single rigid body in $\mathbb{SE}(3)$ as follows:

$$x = [\bar{q}_b, r, \omega, v]^T \tag{13}$$

where $\bar{q}_b = [q_1, q_2, q_3]$ is the vector part of the unit quaternion, $r \in \mathbb{R}^3$ is the position vector, ω the angular velocity, and $v = \dot{r}$ is the linear velocity, all expressed with respect to the inertial frame of reference (also known as hybrid representation in kinematics [18]). The derivative of the state vector can now be written as:

$$\dot{x} = [\dot{\bar{q}}_b, v, \dot{\omega}, \dot{v}]^T \tag{14}$$

The equation of motion for this rigid body in matrix form without gravity can be written as follows [16]:

$$M(q)\ddot{q} + C(q,\dot{q}) = u \tag{15}$$

Here, $M(q) \in \mathbb{R}^{6\times 6}$ is the generalized mass-inertia matrix of the rigid body, $C(q,\dot{q}) \in \mathbb{R}^6$ is the generalized Coriolis and Centrifugal effort, and $u \in \mathbb{R}^6$ is the generalized forces vector. q, $\dot{q}$, and $\ddot{q}$ are the generalized positions, velocities, and accelerations of the system. For a single rigid body, $\ddot{q}$, M, and u can be written as:

$$\ddot{q} = [\dot{\omega}, \dot{v}]^T \tag{16}$$

$$M = \begin{bmatrix} I_b & 0 \\ 0 & mE \end{bmatrix} \tag{17}$$

$$u = [\tau, F]^T \tag{18}$$

Here, $I_{b3\times 3}$ is the inertia matrix, E is a 3×3 identity matrix, m is the mass of the body, and $\tau_{3\times 1}$ and $F_{3\times 1}$ are the torques and Forces applied about the center of mass. The Equation 16 can be evaluated using the Equation 15, Equation 17, and Equation 18 at a fixed point ($\dot{q} = 0$) to obtain the following expression:

$$\ddot{q} = \begin{bmatrix} \dot{\omega} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} I_b{}^{-1}\tau \\ \frac{1}{m}F \end{bmatrix} \tag{19}$$

Using Equation 19 and Equation 11, the derivative of the state vector from Equation 14 can be written as follows:

$$\dot{x} = \left[\frac{1}{2}Q\omega, v, I_b{}^{-1}\tau, \frac{1}{m}F\right]^T \tag{20}$$

We can now take a first order Taylor approximation of Equation 20 about a fixed point (x^*, u^*):

$$\dot{x} \approx f(x^*,u^*) + \left[\frac{\partial f}{\partial x}\right]_{x=x^*,u=u^*}(x - x^*) + \left[\frac{\partial f}{\partial u}\right]_{u=x^*,u=u^*}(u - u^*) \tag{21}$$

Using the stationary fixed point at origin (with quaternion $q_b = [1,0,0,0]^T$), the state and the generalized forces can be written as $x^* = 0_{12\times 1}$ and $u^* = 0_{6\times 1}$ respectively. After evaluating the partial derivatives, the linear system can be written as:

$$\dot{x} = Ax + Bu \tag{22}$$

where:

$$A = \left[\frac{\partial f}{\partial x}\right]_{x=x^*,u=u^*} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & \frac{1}{2}E_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & E_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{23}$$

$$B = \left[\frac{\partial f}{\partial u}\right]_{u=x^*,u=u^*} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \\ I_{b3\times3}^{-1} & 0_{3\times3} \\ 0_{3\times3} & \frac{1}{m}E_{3\times3} \end{bmatrix} \tag{24}$$

The above equations provide a linear system for the dynamics of the single rigid body in $\mathbb{SE}(3)$ parameterized using quaternions. It can be easily verified that the given linear system is controllable. Thus, a Linear Quadratic Regulator (LQR) can be used to control this system about its fixed point. The LQR controller is locally optimal and globally asymptotically stable [3].

3.3 Multi-Body Linearization using Quaternions

The linear system given in the Sub-Section 3.2 can be extended to a free-floating multi-body system with configuration space $\mathbb{SE}(3) \times \mathbb{R}^n$ where n is the number of joints in the system. The state vector for a multi-body system can be written as:

$$x = [\bar{q}_b, r_b, q_m, \omega_b, v_b, \dot{q}_m]^T \tag{25}$$

Here, $\bar{q}_b$ is the vector part of the unit quaternion representing the floating base orientation, r_b is the position vector of the center of mass of the floating base, q_m is the vector of generalized joint positions, ω_b the angular velocity of the floating base, v_b the linear velocity of the floating base, and $\dot{q}_m$ is the generalized joint velocity vector, all expressed with respect to the inertial frame of reference (also known as hybrid representation in kinematics [18]). Similarly, the derivative of the state can be written as:

$$\dot{x} = [\dot{\bar{q}}_b, v_b, \dot{q}_m, \dot{\omega}_b, \dot{v}_b, \ddot{q}_m]^T \tag{26}$$

Using a fixed point for linearization (where $\dot{q} = [\dot{\bar{q}}_b, v_b, \dot{q}_m] = 0$), the equations of motions from Equation 15 can be written as:

$$M(q)\ddot{q} = u \tag{27}$$

$$\ddot{q} = \begin{bmatrix} \dot{\omega}_b \\ \dot{v}_b \\ \ddot{q}_m \end{bmatrix} = M^{-1}u \tag{28}$$

Using Equation 28, Equation 26, and Equation 11, we can now write the state derivative as:

$$\dot{x} = \left[\frac{1}{2}Q\omega_b, v_b, \dot{q}_m, M^{-1}u\right]^T \tag{29}$$

The generalized forces for the free-floating multi-body system can be written as:

$$u = [\tau_b, F_b, \tau_m]^T \tag{30}$$

where, τ_b and F_b are the torques and forces acting on the floating base, and τ_m are the generalized forces acting on the joints. The state of the system about the fixed point can be written as follows:

$$x = [0_{6\times 1}, q_{m_{n\times 1}}, 0_{(6+n)\times 1}]^T_{(12+2n)\times 1} \tag{31}$$

We can then use the Taylor Expansion (from Equation 21) to create a linearized version of the multibody equations of motion about the fixed point. After evaluating the partial derivatives, the linear system matrices are:

$$A = \begin{bmatrix} 0_{3\times 3} & 0_{3\times 3} & 0_{3\times n} & \frac{1}{2}E_{3\times 3} & 0_{3\times 3} & 0_{3\times n} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times n} & 0_{3\times 3} & E_{3\times 3} & 0_{3\times n} \\ 0_{n\times 3} & 0_{n\times 3} & 0_{n\times n} & 0_{n\times 3} & 0_{n\times 3} & E_{n\times n} \\ 0_{(6+n)\times 3} & 0_{(6+n)\times 3} & 0_{(6+n)\times n} & 0_{(6+n)\times 3} & 0_{(6+n)\times 3} & 0_{(6+n)\times n} \end{bmatrix} \tag{32}$$

$$B = \begin{bmatrix} 0_{3\times 3} & 0_{3\times 3} & 0_{3\times n} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times n} \\ 0_{n\times 3} & 0_{n\times 3} & 0_{n\times n} \\ \multicolumn{3}{c}{M^{-1}_{(6+n)\times(6+n)}} \end{bmatrix} \tag{33}$$

Similar to Sub-Section 3.2, it can be easily verified that the linear system derived here is controllable. An LQR controller can provide globally asymptotic stability of such non-linear system (while ignoring contact) i.e. it can provide stability for any initial position and velocity. Furthermore, near the point of linearization i.e. the fixed point, as the linear system is a good approximation of the non-linear dynamics about this point, the same controller acts as an almost optimal controller. Thereby, a single LQR controller showcases global stability and local optimality. Simultaneously satisfying these two properties is an unique aspect for a single controller for a fully-actuated multibody system in $\mathbb{SE}(3) \times \mathbb{R}^n$.

3.4 Additional Bias Terms

Here we demonstrate how to handle additional bias terms in the equations of motion. Post-linearization additional terms which are linear with respect to either position or velocity can be added to the linear system matrices to increase the fidelity of the linear system. We demonstrate this with the use-case of an underwater vehicle. We derive the controller equations for the case of single rigid body underwater vehicle. In such case two extra components could add bias to the system, namely hydrodynamic damping and a restoring effect resulting from the buoyant and gravitational forces. We write the equations of motion of a fully submerged rigid-body with the linear velocity v represented with respect to the body frame, as it is easier to express the damping terms. The equations of motion are expressed as follows

$$M\ddot{q} + C(\dot{q}) + d(\dot{q}) + g(q) = u \tag{34}$$

where $d(\dot{q})$ represents the damping term and $g(q)$ is the restoring effect. To model the hydrodynamic damping we follow a *second order modulus* approximation which was proposed in [19] and adopted later in [20]. This allows us to write the equations of damping as a combination of a linear and non-linear term as follows

$$d(\dot{q}) = D\dot{q} = (D_l + D_n(\dot{q}))\,\dot{q} \tag{35}$$

Here, $D \in \mathbb{R}^{6\times 6}$ is a time-variant damping matrix, $D_l \in \mathbb{R}^{6\times 6}$ is a constant matrix representing the linear drag terms, and $D_n(\dot{q}) \in \mathbb{R}^{6\times 6}$ is a term representing the non-linear drag components described as:

$$D_n(\dot{q}) = D_q \text{diag}(|\omega|, |v|) \tag{36}$$

where $D_q \in \mathbb{R}^{6\times 6}$ is a constant matrix representing the non-linear damping coefficients. The restoring terms are contributed to the effect due to the gravitational pull and the buoyancy forces acting on a fully submerged body. Assuming a fixed-mass and fixed-volume body, the gravitational and buoyant forces can be considered constant vectors f_G and f_B, pointing towards and out of earth's center, respectively. We assume here a neutrally buoyant vehicle ($f_G = -f_B$), which allows us to express the restoring terms as follows

$$G(q) = \begin{bmatrix} r_B \times R^T f_B \\ 0_{3\times 3} \end{bmatrix} \tag{37}$$

where $r_B = [r_x, r_y, r_z]^T$ is the position of the center of buoyancy in the body frame, and $R \in \mathbb{SO}(3)$ is a rotation matrix between the body and inertial frame.

Based on this formulation, we update the matrix A in Equation 23 as a time varying matrix expressed as follows

$$A = \begin{bmatrix} 0_{3\times 3} & 0_{3\times 3} & \frac{1}{2}E_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & E_{3\times 3} \\ G_{lin} & 0_{3\times 3} & D^{\omega} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & D^{v} \end{bmatrix} \tag{38}$$

Here D^{ω} and D^{v} are the upper left and lower right matrices of time-varying damping matrix D in (35), respectively. G_{lin} is the linearization of the restoring term G about $q_b = [1,0,0,0]^T$ given as:

$$G_{lin} = 2b \begin{bmatrix} -r_z & 0 & 0 \\ 0 & -r_z & 0 \\ r_x & r_y & 0 \end{bmatrix} \tag{39}$$

where b is the net buoyancy.

4 Example Applications

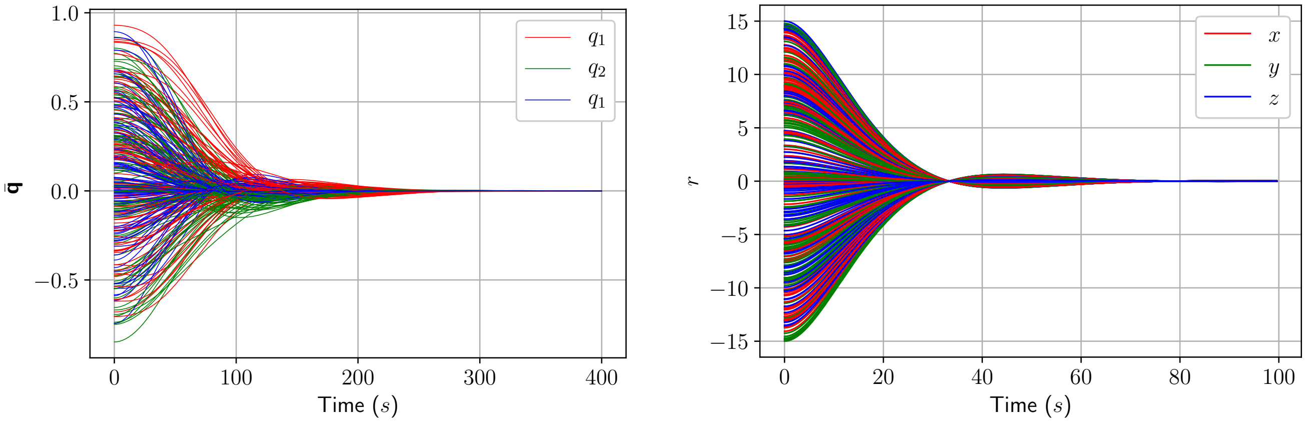
The applicability of the methods derived in Section 3 are shown here using 2 different application areas: Space and Underwater Robotics.

4.1 Space Robotics

In this section, we use linear systems created using quaternion-based lineararization from Section 3 to control two different robotic systems in space. The first system is an asymmetric rigid body whose position-orientation is controlled using the derived LQR controller. The second use-case is the stabilization of a robotic system post capture of a tumbling space debris. All simulations were performed using the Drake software [21].

4.1.1 Control of an Asymmetrical Rigid Body

We use the linear system based LQR to control the asymmetrical rigid body given in the reorientation trajectory optimization example in [1]. The system is linearized about the origin and various random initial conditions are generated. The random initial conditions are used to perform a full non-linear dynamic simulation of the system while being controlled using the LQR controller. For the simulations, the random initial unit quaternion rotation was generated by taking a random unit vector for the axis, and the angle α was randomly sampled between $-180°$ and $180°$. The position vector was generated by sampling randomly between $-15\,\mathrm{m}$ and $15\,\mathrm{m}$ for each of the axis. The gains used for the simulation were $Q = \mathrm{diag}(100,100,100,2,2,2,50,50,50,2,2,2)$ and $R = \mathrm{diag}(1,1,1,1,1,1)$. These gains were selected based on the significance of the errors each of the DoF. The current gains focus on maintaining attitude and angular velocity but different gains can be used depending on the use case. The results following 100 simulations can be seen in Figure 2. Figure 3 shows a few snapshots of the simulation during one of the control maneuvers where the satellite CAD model is adapted from [22].



(a) Quaternions

(b) Positions

Figure 2 Quaternion and Position Output using the Derived $\mathbb{SE}(3)$ LQR Controller. The time span of the position plot is smaller as it converges earlier.

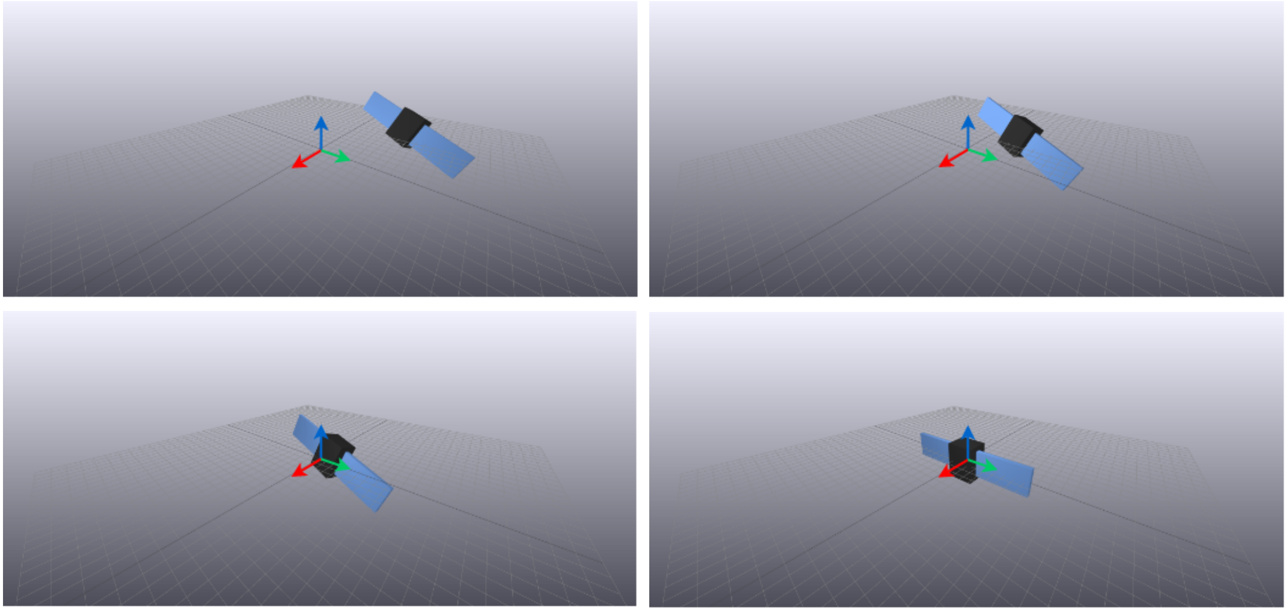
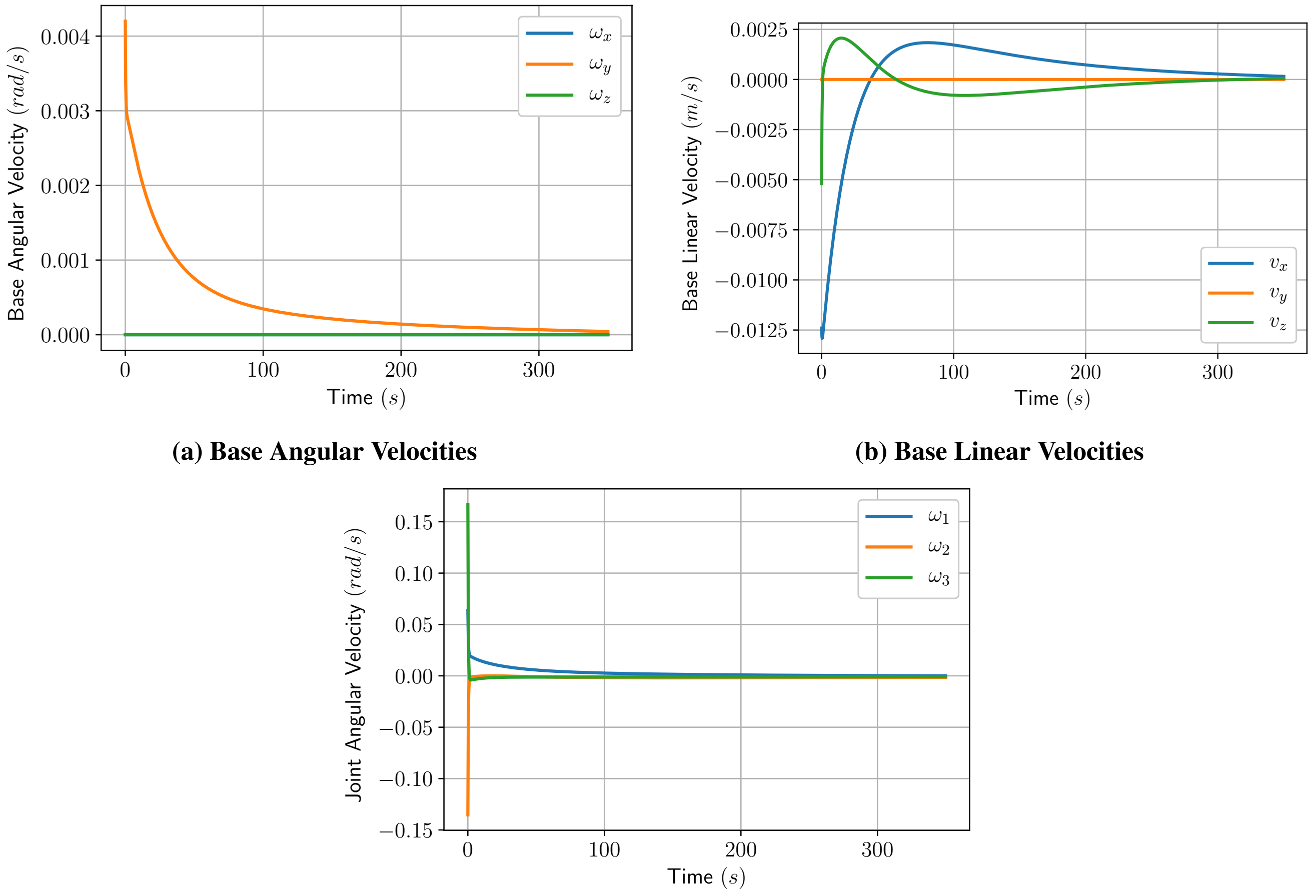


Figure 3 Different snapshots of the Satellite during a control maneuver. The top-left figure shows the initial position, top-right and bottom-left show the transition, and the bottom-right image shows the Satellite at the final state.

4.1.2 Post-Capture Stabilization of Tumbling Space Debris

For post-capture detumble scenario, we consider the case of a chaser spacecraft with a 3 degree of freedom robot arm capturing a tumbling target satellite. Assuming perfect velocity-position synchronisation between the end-effector of the chaser and the grasping point on the target spacecraft, we can assume an ideal capture scenario without any contact forces. Thus, the state before and after the capture are the same. In this way, we can obtain the initial state of the system post-capture by utilizing resolved motion rate control using the Generalized Jacobian Matrix [23] to follow the grasping point trajectory perfectly using the end-effector. For the simulation, consider the target tumbling at a rate of $5\,^\circ\mathrm{s}^{-1}$. The chaser spacecraft has a mass of 100 kg (evenly distributed about a cube of side 2 m), the links have masses 10 kg, 8 kg, and 4 kg and lengths 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 side 0.6 m). We then apply LQR using the linear system derived in Sub-Section 3.3 to control this system so as to achieve a detumbled state with zero velocity. As the focus was on detumbling, high-gains were used in the Q matrix for the velocity components, with special focus on the base rotation rates: $Q_v = \mathrm{diag}(100, 200, 100, 10, 10, 10, 10, 10, 10)$. All the position gains were set as 0.0001 as the final position is not prioritized for the detumbling scenario. The actuation gain matrix R was set as an identity matrix. The results obtained from using the LQR controller for post-capture detumbling of space debris can be seen in Figure 4.



(a) Base Angular Velocities

(b) Base Linear Velocities

(c) Joint Velocities

Figure 4 Resultant Velocities while using the Derived $\mathbb{SE}(3) \times \mathbb{R}^n$ LQR Controller for Detumbling

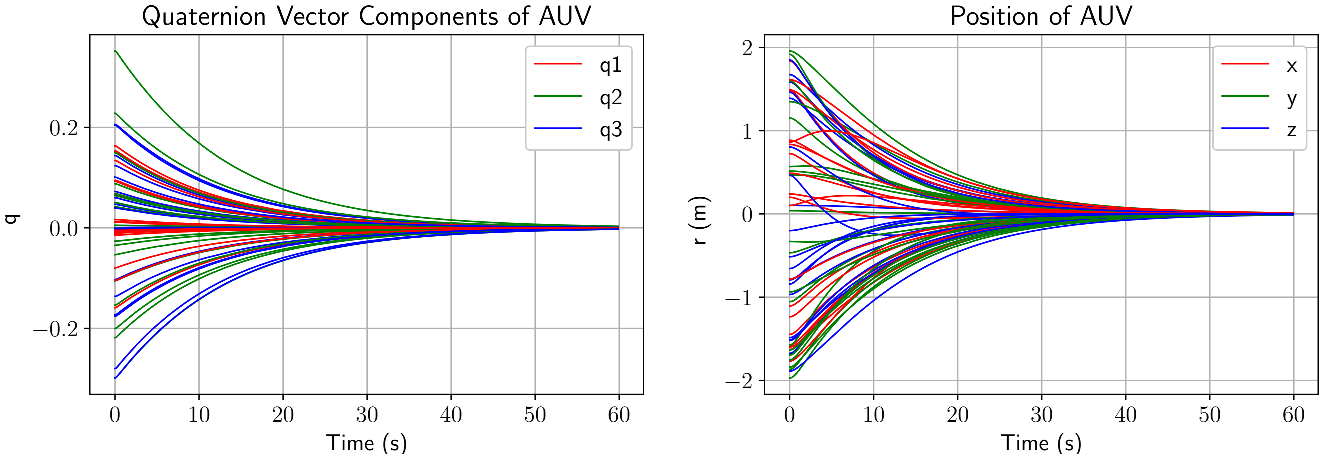
4.2 Autonomous Underwater Vehicles

In this section, we demonstrate the proposed Quaternion-based linearization to control an underwater robotics system. As a study subject, we use a simulated model of the AUV *DeepLeng* [24]. The vehicle can be modeled as a single fully submerged rigid-body.

We test the proposed controller on a simulated vehicle in Gazebo using a plugin *GazeboUnderwater*[1] [25] to simulate the effect of water on an underwater system. We use random initial poses generated in a similar manner to the space robotics application in section 3.2. We run 20 simulations for 60 seconds each using the following gains $Q = \mathrm{diag}(10, 10, 10, 5, 5, 5, 1, 1, 1, 1, 1, 1)1e^2$ and $R = \mathrm{diag}(2, 2, 2, 2, 2, 2)$. The gains focus on maintaining the zero position and orientation. Results showing the vehicle's orienta-

[1] `https://github.com/rock-gazebo/simulation-gazebo_underwater`

tion and position are depicted in figures 5a and 5b, respectively. Figure 1 shows a few snapshots of the simulation during one of the control maneuvers.



(a) Quaternions **(b) Positions**

Figure 5 Quaternions and position output of the simulated AUV using the Derived $\mathbb{SE}(3)$ LQR Controller

5 Conclusion and Future Work

In this paper, we have provided an analytical method for quaternion-based linearization of the equations of motion of free-floating single and multi-body systems. The linear system derived in this paper is fully controllable thus allowing the use of the rich literature of control and analysis methods of linear systems. The utility of this linearization method was demonstrated by utilizing LQR to control the linearized systems in space and underwater robotics domains. Additional bias terms were added to the linear model for underwater robotic systems to increase the fidelity of the linear system. The quaternion-based linear control method shown here allows for a complete control pipeline using quaternions: from trajectory optimization to trajectory stabilization and control. Trajectory stabilization requires a time-varying linearization using the method presented here and will constitute as our next step. Along with this, future work includes adding various actuator models for AUVs in the linearization process and exploring the effects of partial state feedback. These will ensure a quaternion-based pipeline for trajectory generation to trajectory execution on real-systems.

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: FKZ 01IW20004) projects funded by the German Aerospace Center (DLR) with federal funds from the Federal Ministry of Education and Research (BMBF).

References

[1] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, jan 2010. DOI: 10.1137/1.9780898718577.

[2] Brian E. Jackson, Kevin Tracy, and Zachary Manchester. Planning with Attitude. *IEEE Robotics and Automation Letters*, 6(3):5658–5664, 2021. DOI: 10.1109/LRA.2021.3052431.

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

[4] A. Chovancová, T. Fico, P. Hubinský, and F. Duchoň. Comparison of various quaternion-based control methods applied to quadrotor with disturbance observer and position estimator. *Robotics and Autonomous Systems*, 79:87–98, 2015. DOI: 10.1016/j.robot.2016.01.011.

[5] Philipp Foehn and Davide Scaramuzza. Onboard State Dependent LQR for Agile Quadrotors. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 6566–6572, 2018. DOI: 10.1109/ICRA.2018.8460885.

[6] Zhao Shulong, An Honglei, Zhang Daibing, and Shen Lincheng. A new feedback linearization LQR control for attitude of quadrotor. *2014 13th International Conference on Control Automation Robotics and Vision, ICARCV 2014*, 2014(December):1593–1597, 2014. DOI: 10.1109/ICARCV.2014.7064553.

[7] Sean Mason, Ludovic Righetti, and Stefan Schaal. Full dynamics LQR control of a humanoid robot: An experimental study on balancing and squatting. *IEEE-RAS International Conference on Humanoid Robots*, 2015-Febru:374–379, 2015. DOI: 10.1109/HUMANOIDS.2014.7041387.

[8] John Ting Yung Wen and Kenneth Kreutz-Delgado. The Attitude Control Problem. *IEEE Transactions on Automatic Control*, 36(10):1148–1162, 1991. DOI: 10.1109/9.90228.

[9] Robert J. Wallsgrove and Maruthi R. Akella. Globally stabilizing saturated attitude control in the presence of bounded unknown disturbances. *Journal of Guidance, Control, and Dynamics*, 28(5):957–963, 2005. DOI: 10.2514/1.9980.

[10] Russell A. Paielli and Ralph E. Bach. Attitude control with realization of linear error dynamics. *Journal of Guidance, Control, and Dynamics*, 16(1):182–189, 1993. DOI: 10.2514/3.11444.

[11] AWI Mohamed, CM Saaj, A Seddaoui, S Eckersley, et al. Controlling a non-linear space robot using linear controllers. In *5th CEAS Conference on Guidance, Navigation and Control (EuroGNC), April 3-5, 2019, Milan, Italy*, 2019.

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

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

[14] Emil Fresk and George Nikolakopoulos. Full quaternion based attitude control for a quadrotor. In *2013 European control conference (ECC)*, pages 3864–3869. IEEE, 2013.

[15] Elias Reyes-Valeria, Rogerio Enriquez-Caldera, Sergio Camacho-Lara, and Jose Guichard. Lqr control for a quadrotor using unit quaternions: Modeling and simulation. In *CONIELECOMP 2013, 23rd International Conference on Electronics, Communications and Computing*, pages 172–178. IEEE, 2013.

[16] Kevin M. Lynch and Frank Park. *Modern Robotics -Mechanics, Planning, and Control*. Number May. Cambridge University Press, 2017. DOI: 10.1017/9781316661239.

[17] Michael S. Andrle and John L. Crassidis. Geometric integration of quaternions. *AIAA/AAS Astrodynamics Specialist Conference 2012*, pages 1–10, 2012. DOI: 10.2514/6.2012-4421.

[18] Andreas Müller. Screw and lie group theory in multibody kinematics. *Multibody System Dynamics*, Jul 2017. DOI: 10.1007/s11044-017-9582-7.

[19] KK Fedyaevsky and GV Sobolev. Control and stability in ship design. 1964.

[20] Thor I Fossen. *Guidance and control of ocean vehicles*. John Wiley & Sons Inc, 1994.

[21] Russ Tedrake and the Drake Development Team. Drake: Model-based design and verification for robotics, 2019.

[22] Josep Virgili-Llop, DV Drew, Marcello Romano, GV Hobson, BE Wakefield, and WB Roberts. Spacecraft robotics toolkit: an open-source simulator for spacecraft robotic arm dynamic modeling and control. In *6th International Conference on Astrodynamics Tools and Techniques*, 2016.

[23] Yoji Umetani and Kazuya Yoshida. Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix. *IEEE Transactions on Robotics and Automation*, 5(3):303–314, 1989. DOI: 10.1109/70.34766.

[24] Marc Hildebrandt, Sascha Arnold, Philipp Kloss, Bilal Wehbe, and Michael Zipper. From epi-to bathypelagic: Transformation of a compact auv system for long-term deployments. In *2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)(50043)*, pages 1–6. IEEE, 2020.

[25] João Britto, André Conceição, Sylvain Joyeux, and Jan Albiez. Improvements in dynamics simulation for underwater vehicles deployed in gazebo. In *OCEANS 2017-Anchorage*, pages 1–6. IEEE, 2017.