

AUV TRAJECTORY OPTIMIZATION WITH HYDRODYNAMIC FORCES FOR ICY MOON EXPLORATION

Lukas Rust^{1,2}, Shubham Vyas^{2,3}, Bilal Wehbe², and Frank Kirchner^{1,2,3}

¹Fachbereich 03, Universität Bremen, Germany, lrust@uni-bremen.de

²Robotics Innovation Centre (RIC), DFKI GmbH, Germany, {lukas.rust, bila.wehbe, shubham.vyas, frank.kirchner}@dfki.de

³AG Robotik, University of Bremen, Germany

ABSTRACT

To explore oceans on ice-covered moons in the solar system, energy-efficient Autonomous Underwater Vehicles (AUVs) with long ranges must cover enough distance to record and collect enough data. These usually underactuated vehicles are hard to control when performing tasks such as vertical docking or the inspection of vertical walls. This paper introduces a control strategy for *DeepLeng* to navigate in the ice-covered ocean of Jupiter's moon *Europa* and presents simulation results preceding a discussion on what is further needed for robust control during the mission.

1. INTRODUCTION

In recent decades, many missions [6, 4] have been undertaken to investigate Europa, one of Jupiter's moons. Its surface consists of a thick ice crust which covers a deep ocean underneath it. Even today multiple missions like *ESA's JUICE* [9] mission or *Clipper* by *NASA* [11] are prepared to launch to specifically orbit around Europa. It attracts research because it is one of the most promising places in our solar system that could contain life[5].

To inspect the ocean a system was conceptualized at *DFKI Bremen* which could drill through the ice and then deploy an AUV into the ocean for inspection. The so-called *DeepLeng* was built with design restrictions to minimize energy consumption and also spatial requirements to fit into the cargo space[10]. The slender torpedo-like shape minimizes drag during forward locomotion, increasing the range and speed of the AUV, which is necessary to realize long-term navigation missions to explore large oceans like *Europa's*. The manifold advantages of such a design come at the cost of underactuation, thus the incapability to generate any wrench. The recently emerging field for control of underactuated systems offers many techniques, but applying them to the underwater domain demands the extension of the system model by hydrodynamic effects and deals with free-floating platforms and a singularity-free representation of the orientation. The

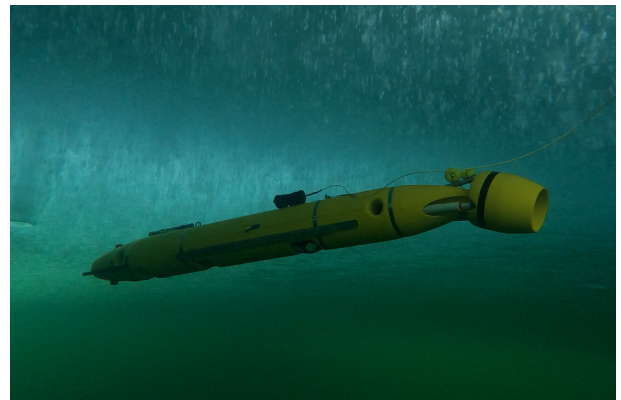


Figure 1: *DeepLeng* AUV demonstrating the vectored thrust during under-ice deployment in Abisko, Sweden

interest to develop and control such AUVs exceeds the exploration of oceans of icy moons and allows the deployment of a new class of underwater robots. This paper contributes a Trajectory Optimization formulation to generate trajectories for chosen agile maneuvers, which explicitly respect the hydrostatic, and hydrodynamic effects and the mechanic constraints. To account for various sources of error, a time-varying Linear Quadratic Regulator is proposed to stabilize the trajectory during execution. The control architecture will show that despite the constraints, maneuvers far outside of the scope of movements, for which the system was originally designed, are possible.

1.1. Related Work

There is extensive research in the field of underactuated systems. Especially for canonical systems, like the cart-pole or the double pendulum, many approaches have been extensively studied. Besides Trajectory Optimization and Linear Quadratic Regulators (LQRs), Model Predictive Control (MPC) [15], Reinforcement Learning Agents (RL) [12] and Proportional Integrative Derivative (PID) [16] controllers have been used.

Usually, those canonical systems are fixed to the world. Underactuated free-floating platforms and their control were investigated to a lesser extent [13, 7]. Trajectory Optimization in the underwater domain has been studied in [2] with a focus on long-distance navigation and not on agile maneuvering. A comparable approach tested on a similar vehicle is presented in [3]. The featured AUV is very similar to *DeepLeng*. Both are only actuated by one large thruster at the rear of the vehicle. Nonlinear and linearized MPC controllers are used to perform hydrodynamics such as the control of an AUV as an inverted pendulum. The controller can explicitly respect the system constraints posed by the maximum angle of the thruster while also respecting the hydrodynamic effects. Examples as sideward movements tested in the paper prove that the controller is capable of performing movements where classical techniques would generate actuation inputs that exceed the constraints of the system.

Due to the high computational cost, MPC can only optimize the control inputs for a restricted horizon and a reduced model. When generating reference trajectories offline or before the execution of the trajectory, one can use a nonlinear model over a longer horizon and then stabilize around the trajectory with the MPC from [3] or other approaches.

2. SYSTEM DYNAMICS

2.1. DeepLeng

DeepLeng has a single thruster at the rear of its body. It is mounted with a pan-tilt unit consisting of a rod and two linear actuators which fix the thruster to the main body and allow to alter the yaw and pitch angle of the thruster to the body. The mechanism offers high stiffness and ro-

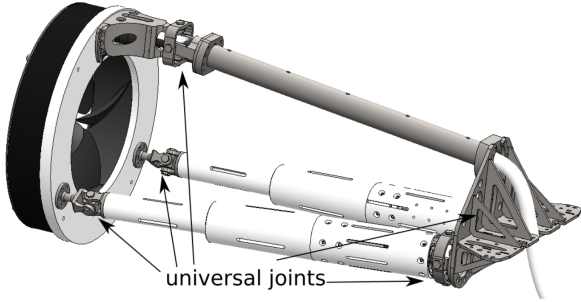


Figure 2: Pan-tilt unit of *DeepLeng*

bustness, but the maximum angular speed of the thruster is relatively slow.

Following the parallel kinematics of this mechanism are not considered. Instead, the yaw and pitch angle of the thruster relative to the main body are used. Solutions to the inverse kinematics $(l_1, l_2) = k(\psi, \phi)$ of this mechanism which maps the thruster angles to the length of the linear actuators are well known and can be computed online. The relation between a wrench ω acting on the

thruster and the generalized forces f acting on the body is given by:

$$f = J^T(s)\omega \quad (1)$$

Where $J^{8 \times 6}$ is the Jacobian of the rigid multibody kinematics and s the generalized positions of the system.

2.2. Dynamics

Computing the full hydrodynamic effects is computationally expensive and not solvable online. To still estimate the force affecting the AUV one has to use simplified models to compute the drag forces.

Fossen [8] models the hydrodynamics as:

$$M\dot{v}C(v)v + D(v)v + r(s) = \tau(s, u) \quad (2)$$

M is the mass matrix consisting of the mass matrix of the vehicle and M_{AM} of the added mass due to the water which must be accelerated with the body. $C(v) = C(v)_M + C_{AM}$ are the Coriolis and centripetal forces of the mass matrix and the added mass.

$D(v)v = D_l v + D_q |v|v$ is the damping term consisting of two coefficients representing linear and quadratic drag terms.

$r(s)$ are the restoring forces. Since we assume a neutrally buoyant vehicle, for the gravitational force and the buoyant force $|f_g| = |f_b|$ holds, it only has a torque component:

$$c_g \times f_g - c_b \times f_b \quad (3)$$

c_g being the center of gravity and c_b the center of buoyancy.

3. CONTROL

3.1. Trajectory Optimization

The decision variables of the Trajectory Optimization are generated by using Direct Transcription. Defining a number of time steps $T \in \mathbb{N}$, we define a set of states $X = \{x_0, \dots, x_{T+1}\}$ and control inputs $U = \{u_0, \dots, u_T\}$. To allow the solver to find a time optimal solution, we define a decision variable h_Δ representing the time step, and the total length of the trajectory as $D = Th_\Delta$. The decision variables are $Z = \{X, U, d_\Delta\}$. The state derivative is defined as $\dot{s}_t = f(s_t, u_t)$, where $f(s_t, u_t)$ is the nonlinear state derivative at a state s_t and an input u_t .

States and the input are defined as:

$$s = (q_w, q_x, q_y, q_z, x, y, z, \psi, \phi)^T \quad (4)$$

$$v = (w_x, w_y, w_z, \dot{x}, \dot{y}, \dot{z}, \dot{\psi}, \dot{\phi})^T \quad (5)$$

$$s_q = (q_w, q_x, q_y, q_z)^T \quad (6)$$

$$s_p = (x, y, z)^T \quad (7)$$

$$x = \begin{pmatrix} s \\ v \end{pmatrix} \quad (8)$$

$$u = (f, \tau_\psi, \tau_\phi)^T \quad (9)$$

The generalized positions s consist of a quaternion and the three-dimensional position, the yaw angle ψ and the pitch angle ϕ of the thruster. The corresponding velocity v consists of three angular velocities around the three principal axes and the derivatives of the position and the thruster angles. s_q is the quaternion part of the state vector, and s_p is the position part. v is the derivative of s , except that the derivative of the orientation is given as angular velocities instead of derivatives of the quaternion. x is the state vector of the system. The input vector contains the force applied by the thruster, whose direction and thus its resulting wrench on the main body depends on ψ and ϕ . τ_ψ and τ_ϕ are the torques applied on the virtual joints through which the thruster is mounted to the main body. The optimization problem:

$$\min_{S, V, u} \sum_{t=0}^{T+1} (s_p^t - s_p^f)^2 + \sum_{t=0}^T u_t^2 + (Td_\Delta)^2 \quad (10)$$

Subject to:

$$F_{min} \leq f \leq F_{max} \quad (11)$$

$$h_{\Delta_{min}} \leq h_\Delta \leq h_{\Delta_{max}} \quad (12)$$

$$0.999 \leq \|s_q\| \leq 1.001 \quad (13)$$

$$\dot{s}_q = \begin{pmatrix} 0 \\ w_x \\ w_y \\ w_z \end{pmatrix} * s_q \quad (14)$$

$$s_{t+1} = s_t + h_\Delta \dot{s}_{t+1} \quad (15)$$

$$v_{t+1} = v_t + h_\Delta \dot{v}_{t+1} \quad (16)$$

$$M\dot{v}_t + Cv + D(v)v = J^T \begin{pmatrix} 0 \\ \tau_\phi \\ \tau_\psi \\ f \\ 0 \\ 0 \end{pmatrix} \quad (17)$$

$$h_{\Delta_{t+1}} = h_{\Delta_t} \quad (18)$$

$$\psi_{min} \leq \psi \leq \psi_{max} \quad (19)$$

$$\dot{\psi}_{min} \leq \dot{\psi} \leq \dot{\psi}_{max} \quad (20)$$

$$\phi_{min} \leq \phi \leq \phi_{max} \quad (21)$$

$$\dot{\phi}_{min} \leq \dot{\phi} \leq \dot{\phi}_{max} \quad (22)$$

$$s_{t=0} = s_{init} \quad (23)$$

$$v_{t=0} = v_{init} \quad (24)$$

$$\dot{v}_{t=0} = \dot{v}_{init} \quad (25)$$

$$s_{t=T} = s_f \quad (26)$$

$$v_{t=T} = v_f \quad (27)$$

$$\dot{v}_{t=T} = \dot{v}_f \quad (28)$$

The costs are defined in Equation 10 as the positional error between the position at any given timestamp t and the desired final position s_p^f , the quadratic input and the total time of the trajectory squared. Equation 11 enforces that the input force does not exceed the maximum force of the thruster. The timestamp h_Δ in Equation 12 must be limited, otherwise the solver fails to find a solution. Equation 13 ensures that s_q has unit length and Equation 14 yields the derivative of the quaternion part of the state by hamiltonian multiplication of the angular part of v and the orientation.

Equation 15 and Equation 16 determine the discrete integration of the state and the velocities. The acceleration is chosen according to the dynamics of the system by the constraint in Equation 17 according to the equation of motion from the previous chapter. Equation 19, Equation 20, Equation 21 and Equation 22 define the maximum angles and angular speeds of the thruster. This constraint allows us to ignore the parallel kinematics and work with joints in the simulation even without identifying the actual mass and inertia of the thruster, as the angular velocities and the angle can be used as inputs to the thruster on the physical system. The latter equations define that the first and final states must equal a user-defined target and goal.

3.2. Trajectory Stabilization

To compensate for errors in the model, unknown environmental forces and errors in the state estimation, a time-varying LQR stabilizes the AUV around the reference trajectory. The state of the system and the control inputs are thus given by:

$$\hat{x} = x - x^* \quad (29)$$

$$\hat{u} = u - u^* \quad (30)$$

where x^* and u^* are the optimal state and input given by the trajectory. The linearization is defined as a first-order Taylor Expansion

$$f(x, u) \approx f(x^*, u^*) + \frac{\delta f(s, u)}{\delta s} \hat{x} + \frac{\delta f(s, u)}{\delta u} \hat{u} \quad (31)$$

This equation can be written in state space form as

$$\dot{f}(x, u) = Ax + Bu \quad (32)$$

where the A matrix is given as

$$A(x) = \begin{pmatrix} 0^{4 \times 9} & \frac{\delta q}{\delta w}^{4 \times 3} & 0^{4 \times 5} \\ 0^{5 \times 9} & 0^{5 \times 3} & \mathbb{I}^{5 \times 5} \\ \mu^{8 \times 9} & & C(s, v)^{8 \times 8} \end{pmatrix} \quad (33)$$

where μ is:

$$\mu = M^{-1} \left(\frac{\delta r(s)}{\delta s} + \frac{\delta B}{\delta s} + \frac{\delta C(s, v)}{\delta s} \right) \quad (34)$$

and the input matrix as:

$$B = \begin{pmatrix} 0^{9 \times 1} & 0^{9 \times 1} & 0^{9 \times 1} \\ M^{-1}J^T & M^{-1} \begin{pmatrix} 0^{6 \times 1} \\ 1 \\ 0 \end{pmatrix} & M^{-1} \begin{pmatrix} 0^{7 \times 1} \\ 1 \end{pmatrix} \end{pmatrix} \quad (35)$$

Calculating these matrices for all the timestamps where each state is associated with a point in time and then interpolating yields the time-varying matrices $A(t)$ and $B(t)$. Since there is no restriction for LQRs that the state and input matrices must be time-invariant, a time-varying LQR can be computed from $A(t)$ and $B(t)$.

Stability describes the dynamics of a system over an infinite time horizon, but our trajectory is only defined over a finite horizon. To stabilize around the trajectory we first have to compute an infinite horizon LQR with the state and input matrix at the last point of the trajectory. Solving the algebraic Riccati equation:

$$0 = S^\infty A + A^T S^\infty - S^\infty B R^{-1} B^T S^\infty + Q \quad (36)$$

yields the cost to go matrix S and the control law:

$$u^* = -R^{-1} B^T S^\infty x = -K^\infty x \quad (37)$$

The finite horizon LQR can be obtained by solving the differential Riccati Equation

$$-\dot{S}(t) = S(t)A + A^T S(t) - S(t)B R^{-1} B^T S(t) + Q \quad (38)$$

with the terminal condition:

$$S(Td_\Delta) = S^\infty \quad (39)$$

and the control law:

$$u^* = -R^{-1} B^T S(t)x = -Kx \quad (40)$$

The control law for the complete trajectory can then be written as:

$$K^f = \begin{cases} K & t \leq Td_\Delta \\ K^\infty & \text{else} \end{cases} \quad (41)$$

In this formulation, it can be clearly seen that the linear system is not controllable due to the quaternions as 4 numbers are used to represent 3 rotational DoFs. Similar to [1], the unit length of the quaternion is therefore enforced directly by reducing the state vector of its first entry. The state size is reduced by one compared to the formulation above. Instead, the real part of the quaternion is calculated as:

$$q_w = 1 - \sqrt{q_x^2 + q_y^2 + q_z^2} \quad (42)$$

4. RESULTS

This section will first introduce three trajectories yielded from the Trajectory Optimization. They represent movements close to the boundaries of the mechanical constraints of *DeepLeng* and they heavily rely on hydrodynamics. The time-varying LQR is tested by executing

the reference trajectories in a simulation with the same parameters for hydrodynamic damping used in the Trajectory Optimization.

4.1. Trajectories

The following three trajectories are the case study to evaluate the control framework: (1) Polebalancing, (2) Quarterhelix and (3) Steep elevation. Throughout these cases, we select the maximum and minimum values of the thruster $F_{min} = -70$ N and $F_{max} = 70$ N.

During the pole balancing maneuver, the vehicle swings up and holds an orientation close to 90° . The maneuver is of interest for vertical docking or the inspection of vertical obstacles. It is also similar to the inverted pendulum system. Figure 3 shows the reference position and ori-

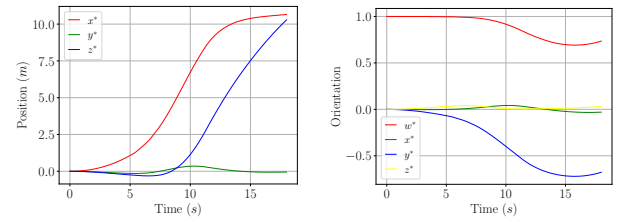


Figure 3: The position and orientation of the pole balancing trajectory

entation of the polebalancing trajectory. The force limit defined in Equation 11 was increased for this trajectory, otherwise the restoring torque, which reaches its maximum at a vertical pose, can not be negated by the thrust force. There is still a margin in the maximum angle and angular velocity of the thrusters to allow the time-varying LQR to stabilize around the trajectory.

Moving upwards or downwards in a helix is of interest for reaching the seabed of the surface whilst staying in a limited area. The maneuver can be subdivided into quarterhelices, where only a curve with a total rotation of 90° is considered. A maneuver of arbitrary duration and range can be easily synthesized by concatenating multiple quarterhelices. Figure 4 shows the trajectories reference position and orientation.

For a monotonic elevation in a tight space, hence with

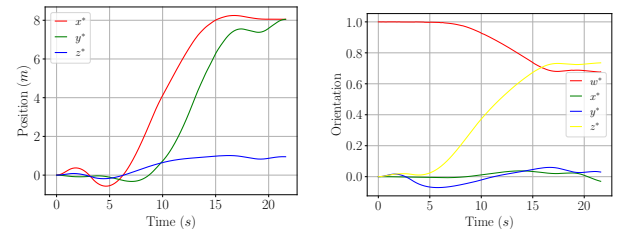


Figure 4: The position and orientation of the quarterhelix trajectory

constraints in the x and y direction of the AUV, the steep elevation trajectory in Figure 5 functions as a reference.

In emergency situations, it is a fast way to reach the surface.

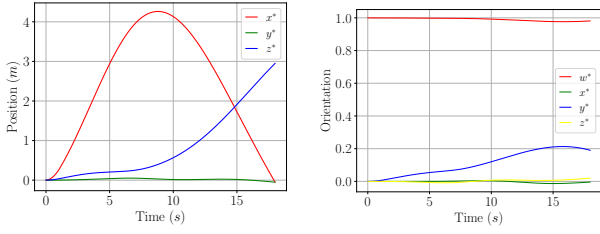


Figure 5: The position and orientation of the steep elevation trajectory

4.2. Simulation

The combined framework of the reference trajectories and the additional time-varying LQR is tested in this simulation. Note that the error passed to the time-varying LQR is not the state of the system, but its difference to the expected state from the reference trajectory. The reference input of the trajectory is subtracted as well.

All tests were conducted using the *Drake* simulator. The hydrodynamics were incorporated by implementing a custom module that applies external forces to the AUV. It uses the formulation from the System Dynamics chapter. Figure 6 shows the position, orientation, the thruster

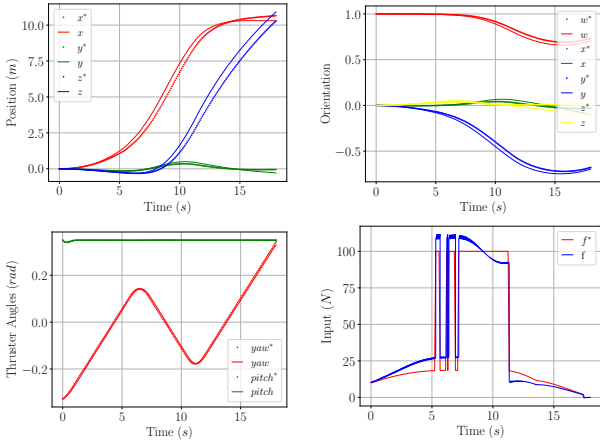


Figure 6: The position(top left), orientation(top right), thruster angles(bottom left) and force input(bottom right) during execution of the pole balancing trajectory

angles and the force input in comparison to the reference trajectories during the execution of the polebalancing trajectory. Figure 7 shows the respective plot for the quarterhelix trajectory and Figure 8 for the steep elevation trajectory.

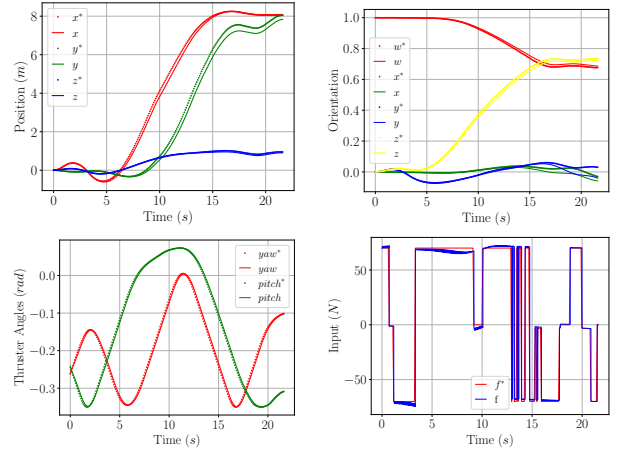


Figure 7: The position(top left), orientation(top right), thruster angles(bottom left) and force input(bottom right) during execution of the quarterhelix trajectory

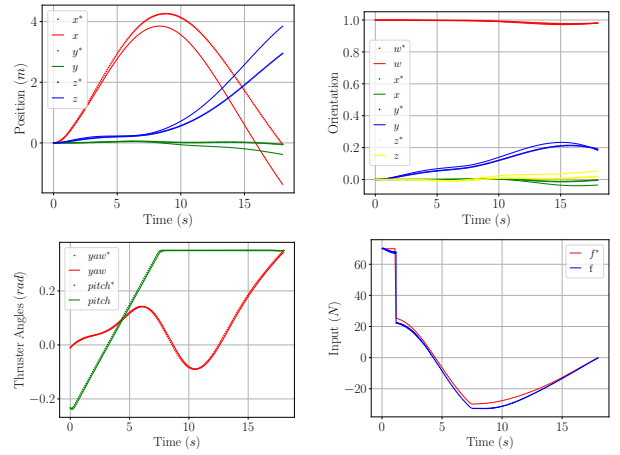


Figure 8: The position(top left), orientation(top right), thruster angles(bottom left) and force input(bottom right) during execution of the steep elevation trajectory

5. DISCUSSION

The system matrix A of the time-varying LQR, which models the dynamics of the system, does not contain the hydrodynamic terms that are applied in the Trajectory Optimization and the simulation. Updating it with the additional bias terms would enhance the controller's performance.

The representation of the *DeepLeng* is generated by parsing a URDF. The modeling of added mass as proposed by [8] is not natively supported in URDF, since that would require the possibility to specify masses for each axis independently. For the above simulation, an average of the added mass was added to the vehicle's mass in the URDF to approximate the dynamics.

[14] describes how to synthesize a controller that is proven to stabilize around the trajectory. Since stability is considered as a statement when time goes to infinity, not only a finite horizon LQR over the time interval of the tra-

jectory is needed but also an infinite horizon LQR which stabilizes around the final pose of the trajectory. The optimal cost-to-go function S of the latter infinite horizon LQR is passed into the finite horizon LQR as a cost function for the final state. In the simulation of this paper, this leads to the failure of the calculation of the finite horizon LQR. Better tuning of the Q and R matrices of the infinite horizon LQR may lead to a cost-to-go matrix S with smaller values which allows the successful computation of the finite horizon LQR.

Till the submission of the paper tuning the time-varying LQR lead to better performance. To the author's assessment, the tuning is still not optimal, and further tuning could improve the performance.

6. CONCLUSION

This paper presented a Trajectory Optimization formulation for an underactuated AUV and a time-varying LQR to stabilize around the reference trajectories. Simulations showed the capability of the controller to execute representative trajectories.

For icy moon exploration, the proposed control architecture will be able to generate and stabilize around nominal trajectories given a precise model, especially of the hydrodynamics. Changes in temperature and alterations in the composition of the liquid have a severe influence on the hydrodynamic parameters. Since the exact properties of the ocean on *Europa* are unknown and due to the controller's dependency on a good model, model identification on Earth will not provide sufficient results.

Adding an online model adaptation could allow the system to correct the errors in the model identification and result in robustness toward unknown dynamics for ocean exploration.

ACKNOWLEDGEMENTS

This work was supported by the project TRIPLE-GNC which received funding from the German Ministry for Economic Affairs and Climate Action (BMWK), FKZ: 50NA2306C. The second author would like to acknowledge the support of M-RoCk (Grant No.: FKZ 01IW21002) and AAPLE (Grant Number: 50WK2275) and the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 813644.

REFERENCES

- [1] Quaternion based lqr for free-floating robots without gravity. In *Proceedings of the 2022 CEAS EuroGNC conference*. Berlin, Germany, May 2022. CEAS-GNC-2022-024.
- [2] Miguel Aguiar, João Borges de Sousa, João Miguel Dias, Jorge Estrela da Silva, Renato Mendes, and Américo S Ribeiro. Trajectory optimization for underwater vehicles in time-varying ocean flows. In *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, pages 1–6. IEEE, 2018.
- [3] Sriharsha Bhat and Ivan Stenius. Controlling an underactuated auv as an inverted pendulum using nonlinear model predictive control and behavior trees. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12261–12267. IEEE, 2023.
- [4] Michel Blanc, Yann Alibert, Nicolas André, Sushil Atreya, Reta Beebe, Willy Benz, Scott J Bolton, Angioletta Coradini, Athena Coustenis, Véronique Dehant, et al. Laplace: A mission to europa and the jupiter system for esa's cosmic vision programme. *Experimental Astronomy*, 23:849–892, 2009.
- [5] Christopher F Chyba. Energy for microbial life on europa. *Nature*, 403(6768):381–382, 2000.
- [6] K Clark, G Tan-Wang, J Boldt, R Greeley, I Jun, R Lock, J Ludwinski, R Pappalardo, T Van Houten, and T Yan. Return to europa: overview of the jupiter europa orbiter mission. In *2009 IEEE Aerospace conference*, pages 1–20. IEEE, 2009.
- [7] Olav Egeland and Kristin Y Pettersen. Free-floating robotic systems. *Control problems in robotics and automation*, pages 119–134, 2005.
- [8] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.
- [9] Olivier Grasset, MK Dougherty, A Coustenis, EJ Bunce, C Erd, D Titov, M Blanc, A Coates, P Drossart, LN Fletcher, et al. Jupiter icy moons explorer (juice): An esa mission to orbit ganymede and to characterise the jupiter system. *Planetary and Space Science*, 78:1–21, 2013.
- [10] Marc Hildebrandt, Jan Albiez, Martin Fritsche, Jens Hilljegerdes, Philipp Kloss, Marius Wirtz, and Frank Kirchner. Design of an autonomous under-ice exploration system. In *2013 OCEANS - San Diego*, pages 1–6. doi: 10.23919/OCEANS.2013.6741164. ISSN: 0197-7385.
- [11] Samuel M Howell and Robert T Pappalardo. Nasa's europa clipper—a mission to a potentially habitable ocean world. *Nature Communications*, 11(1):1311, 2020.
- [12] Savinay Nagendra, Nikhil Podila, Rashmi Ugarakhod, and Koshy George. Comparison of reinforcement learning algorithms applied to the cart-pole problem. In *2017 international conference on advances in computing, communications and informatics (ICACCI)*, pages 26–32. IEEE, 2017.
- [13] Hai-Long Pei and Yangsheng Xu. Control of underactuated free floating robots in space. In *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190)*, volume 2, pages 1364–1369. IEEE, 1998.

- [14] Russ Tedrake. Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation. *Course Notes for MIT*, 6, 2016.
- [15] Nolan Wagener, Ching-An Cheng, Jacob Sacks, and Byron Boots. An online learning approach to model predictive control. *arXiv preprint arXiv:1902.08967*, 2019.
- [16] Anil Kumar Yadav, Prerna Gaur, AP Mittal, and Masood Anzar. Comparative analysis of various control techniques for inverted pendulum. In *India International Conference on Power Electronics 2010 (IICPE2010)*, pages 1–6. IEEE, 2011.