LQR FOR FREE-FLOATING ROBOTS: THEORY AND EXPERIMENTS

Shubham Vyas[1], **Lasse Shala**[1], **and Anton Bredenbeck**[2]

[1]*Robotics Innovation Center, DFKI GmbH, Bremen, Germany, Shuhbam.Vyas@dfki.de*
[2]*TU Delft, Delft, Netherlands*

ABSTRACT

Space robots have been suggested as a prime candidate for On-Orbit Servicing (OOS) and Active Debris Removal (ADR) missions. In this paper, we present the results of employing LQR-based controllers for various free-floating robotic systems. LQR-based controllers have been used frequently as they provide an optimal controller for linear systems. Previous work has shown that the LQR controller for the linearized equations of motion for free-floating robots without gravity is globally asymptotically stable and locally optimal. The linearized equations of motion for 3 different systems are presented along with results from experiments for fixed-point stabilization and trajectory tracking. These systems vary between having continuous actuation using propellers, and binary-pulsed thrusters, along with either a single floating rigid body or a multi-body floating system. LQR controllers allow for trajectory tracking during different phases of OOS or ADR missions. Along with trajectory stabilization, recent works have demonstrated the estimation of the Region of Attraction (RoA) of such controllers for trajectory stabilization. This can be used in the future for sequential controller composition of controllers to guarantee stability through phase transitions. Furthermore, the estimated RoA allows for quick go/no-go decision-making during operations when unaccounted/unmodelled disturbances are observed.

Key words: Free-Floating Robotics; Space Robotics; Orbital Robotics; Free-Floating Control; LQR; Experimental Validation.

1. INTRODUCTION

As Space Debris becomes a larger problem [13], robotic Active Debris Removal (ADR) has been suggested and researched as a method to remove space debris [11]. Along with their ADR capabilities, robots can also be used for On-Orbit Servicing (OOS) [7]. Control of robots with a free-floating base is difficult due to the inherent kino-dynamic coupling in the system as the reaction forces from the manipulator affect the motion of the base and vice-versa. This coupling requires kino-dynamic planning which is more difficult in contrast to traditional fixed base manipulators on Earth. Due to this, research has been carried out on various control methods for robots with free-floating bases for space applications, some of which, the reader can see in [5, 10, 15, 18, 20, 23, 24]. However, due to the free-floating nature of the robotic system, testing these methods on Earth is challenging. Thus, only a few of the methods in literature have been experimentally verified either in space or on a free-floating testbed. One of the most common ways to experimentally test and validate free-floating control algorithms is by using air-bearing testbeds on flat floors such as [4, 8, 19, 22, 24]. This paper provides a report on the use of linear controllers for different types of free-floating systems, such as single or multibody systems and with various actuation modalities such as continuous actuation or binary actuation.

2. BACKGROUND

In this section, we provide a short introduction to the dynamics of free-floating systems due to space constraints. The reader is referred to the literature for a thorough treatment of the topics referred here: [1, 14, 26, 28]. While the dynamics given below are applicable for all free-floating robotic systems, specific adaptations can be made for particular systems which are elaborated in Section 3. The generalized dynamics for free-floating systems without gravity can be written as:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{u} \tag{1}$$

$\boldsymbol{q}$ represents the generalized positions of the system. $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{6+n \times 6+n}$ is the generalized mass-inertia matrix of the system, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{6+n}$ is the generalized Coriolis and Centrifugal effort, and $\boldsymbol{u} \in \mathbb{R}^{6+n}$ is the vector of generalized forces.

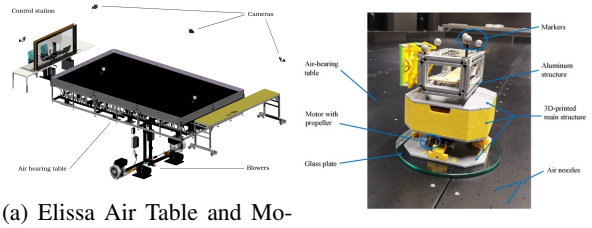
For each of the tests, trajectory optimization was performed using the direct transcription method [21] and the control was performed using a Time-Varying Linear Quadratic Regulator (TVQLR) [21]. For a more in-depth review of the methods used for the experiments in this paper, the reader is referred to the following work: [1, 4, 12, 26].

3. SYSTEM DESCRIPTION

In this section, the various free-floating test systems are described along with the variations in actuation modalities and the adaptations of the modeling and control methods for the specific systems.

3.1. Elissa

Elissa is a planar $3DoF$ free-floating experimental testbed at the Institute for Space Systems at TU Braunschweig [22]. Unlike the more common air-bearing testbeds (such as [4, 9, 16, 17]) where the floating surface is fully solid and the air-bearings are placed along with a gas supply on the free-flying vehicle, the *Elissa* testbed is more akin to a typical air-hockey table where the air required for floating of the spacecraft under test is provided from a large number of small holes on the surface of the table. This allows for larger untethered experiment run times as the onboard air supply capacity is no more a limitation. The filtered air for *Elissa* is supplied using blowers situated in a different room to reduce noise during experiments and allow for continuous operation. The advantage of such a system is that it allows for relatively long-duration experiments. However, the setup does not permit heavier test payloads as the surface area required for the base becomes impractically large. The test spacecraft used for this work weighed around $\approx 4.26\,\mathrm{kg}$. The testbed consists of an air-bearing table, one or more free-flyers, and a motion capture system as seen in Figure 1. The free-flyers consist of Raspberry Pi 3 as the onboard computer and communicate to the lab computer via WiFi using the Robot Operating System (*ROS Noetic*) as the communication middleware. For the experiments reported in this paper, the free-flyer was operated in hardware-in-loop configuration with all the computations performed on the lab computer and the actuation commands relayed to the onboard computer which then executes them and sends back the feedback data. Each free-flyer has eight propellers for actuation operated using brushless DC motors. For state feedback, a motion capture system was used.



(a) Elissa Air Table and Motion Capture System

(b) Elissa Free-Flyer

Figure 1: Elissa Experimental Testbed. From [29]

Each free-flyer is modeled as a single rigid body with a mass $m = 4.26\,\mathrm{kg}$ and the moment of inertia about the single rotation axis $I_{zz} = 0.064\mathrm{kg\,m^2}$. The state vector $\mathbf{x} \in \mathbb{R}^6$ of the system can be written as:

$$\mathbf{x} = [\mathbf{q} \quad \dot{\mathbf{q}}]^T = [x \quad y \quad \theta_z \quad \dot{x} \quad \dot{y} \quad \omega_z]^T \tag{2}$$

here, $\mathbf{q}$ and $\dot{\mathbf{q}}$ are the generalized positions of the system, with x and y being the positions about x- and y-axis, and θ_z being the rotation about the z-axis. Similarly, $\dot{x}$, $\dot{y}$, and ω_z, are the velocities about the x- and y-axis and rotation velocity about the z-axis respectively. The control input $\mathbf{u} \in \mathbb{R}^3$ can be written as $\mathbf{u} = [f_x \quad f_y \quad \tau_z]^T$ where the terms are forces in x and y direction and torque about z-axis respectively.

For the free-flyer system, the dynamics can be written as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3}$$

where $\mathbf{A}$ and $\mathbf{B}$ can be written as:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{I} \end{bmatrix} \tag{5}$$

We can now use the methods for trajectory optimization and control given in Section 2 for synthesizing nearly circular trajectories for this system for an inspection-like scenario and then realize it on the experimental testbed using TVLQR. For the trajectory optimization, a circular trajectory was realized using intermediate points as constraints along with the free-flying facing the center of the circle at all times during the circular trajectory. The trajectory optimization problem can be written as:

$$\begin{aligned} \min_{\mathbf{x}(t),\mathbf{u}(t)} \quad & \int_{t_0}^{t_f} (W_t \Delta t + \mathbf{W}_u \mathbf{u})\, dt \\ \text{s.t.} \quad & \dot{\mathbf{x}(t)} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ & \mathbf{x}\left(t_0\right) = \mathbf{x}_0 \\ & \mathbf{x}\left(t_{circ}\right) = \mathbf{x}_{circ} \\ & \mathbf{x}\left(t_f\right) = \mathbf{x}_f \\ & \mathbf{u}_{\min} \leq \mathbf{u}(t) \leq \mathbf{u}_{\max} \end{aligned} \tag{6}$$

Here, W_t and $\mathbf{W_u}$ refer to the time and actuation cost weights respectively. This trajectory optimization problem's goal is to minimize the time and actuation used

while respecting the linear dynamics and executing a nearly circular trajectory enforced by $\mathbf{x}_{circ}$. This can now be transcribed into a non-linear program which is solved using *SNOPT* [6] solver:

$$\begin{aligned}
\min_{\mathbf{x}_k, \mathbf{u}_k} \quad & \sum \Delta_{t,k} \left(W_t + \mathbf{u}_k^T \mathbf{W}_u \mathbf{u}_k \right) \\
\text{s.t.} \quad & \mathbf{q}_{k+1} = \mathbf{q}_k + \Delta t_k \dot{\mathbf{q}}_k \\
& \dot{\mathbf{q}}_{k+1} = \dot{\mathbf{q}}_k + \Delta t_k \ddot{\mathbf{q}}_k \\
& \ddot{\mathbf{q}}_k - \mathbf{B}\mathbf{u}_k = 0 \\
& \Delta t_{k+1} = \Delta t_k \\
& \mathbf{q}_{30} = \begin{bmatrix} 3 & 1 & \frac{\pi}{2} \end{bmatrix}^T \\
& \mathbf{q}_{50} = \begin{bmatrix} 4 & 2 & \pi \end{bmatrix}^T \\
& \mathbf{q}_{70} = \begin{bmatrix} 3 & 3 & \frac{3\pi}{2} \end{bmatrix}^T \\
& \mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max}
\end{aligned} \tag{7}$$

This results in a nearly circular trajectory as can be seen in Section 4, Figure 6. To run the controller to follow the trajectory on the real system, a control architecture as shown in Figure 2 was implemented.

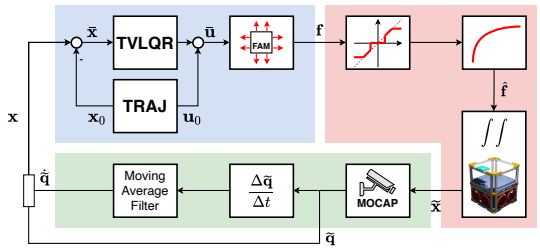


Figure 2: Closed Loop Control Architecture for Elissa. The control-related components are shown in *blue*, the free-flyer plant in *red*, and the state estimation in *green*

The controller's output forces $\mathbf{u}$ are mapped to the propeller actuators using the *Force Allocation Module (FAM)*. The closed-loop control frequency of the system is $\approx$ 100Hz. The free-flyer's propeller has a lower and upper force limit due to aerodynamics at the lower RPM and RPM limit at the upper bound. These are handled by a saturation block with a headband that limits the output from the TVLQR controller. For trajectory optimization, the bounds are handled respectively using appropriate $\mathbf{u}_{min}$ and $\mathbf{u}_{max}$ after mapping the force limits from the propellers to the generalized forces that can be applied to the center of mass of the free-flyer model. The propellers are controlled using an *Electronic Speed Controller (ESC)* and identification was performed to map the force generated for each commanded RPM. For state feedback, *Optitrack* motion capture system was used to accurately obtain the free-flyer's position $\tilde{\mathbf{q}}$. The velocity, $\dot{\tilde{\mathbf{q}}}$, was obtained by numerical differentiation of the position followed by a moving averaging filter. An in-depth report on *Elissa's* software framework can be found in [12].

The 8 propellers on the free-flyer assert a generalized force on the body which can be defined using the following mapping matrix $\mathbf{T} \in \mathbb{R}^{3\times 8}$:

$$\begin{aligned}
\mathbf{u} &= \mathbf{T}.\mathbf{f} \\
&= \begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ a & -a & -a & a & a & -a & -a & a \end{bmatrix} \begin{bmatrix} f_{t_1} \\ f_{t_2} \\ \vdots \\ f_{t_8} \end{bmatrix}
\end{aligned} \tag{8}$$

here, a denotes the propeller's leverage arm w.r.t to the free-flyer's center of mass. To map the actuation output $\mathbf{u}$ from the controller to the propellers, $\mathbf{T}$ has to be inverted. For this, the Moore–Penrose inverse or the *pseudoinverse* of the $\mathbf{T}$ matrix $\mathbf{T}^+$ is used. From this, we get the mapping matrix. However, as the propellers can only provide thrust in one direction, all negative entries in $\mathbf{T}^+$ are removed and the positive entries are multiplied by a factor of 2 [30]. The final mapping matrix $\bar{\mathbf{T}} \in \mathbb{R}^{8\times 3}$ can be written as ([30]):

$$\bar{\mathbf{T}} \quad \text{where} \quad \bar{T}_{ij} = \begin{cases} 2T_{ij}^+ & \text{for} \quad T_{ij}^+ > 0 \\ 0 & \text{for} \quad T_{ij}^+ \leq 0 \end{cases} \tag{9}$$

3.2. REACSA

REACSA is a free-floating test setup at the Orbital Robotics and GNC Lab (ORGL) [31] at ESA-ESTEC. It consists of a platform floating with the use of air bearings on an $5\,\mathrm{m} \times 9\,\mathrm{m}$ epoxy flat floor. The REACSA test platform consists of three components: ACROBAT which provides the floating base with 3 air bearings, SATSIM which provides the air tanks and 8 thrusters, and RECAP which provides the reaction wheel. Combined, these three platforms create REACSA (see Figure 3 for a schematic and left of Figure 5 for the real system) which is then used for experiments for free-floating satellite control. The platform weighs $\approx 220\,\mathrm{kg}$ and thus, is one of the largest free-floating testbeds available in Europe. In-depth information about the platform can be found in [2–4].

The propulsion system in REACSA consists of 8 thrusters mounted as 4 counter-facing pairs which can be used to apply forces and torques on the system. Ideal thrust is assumed as long as the minimum on-time of the thrust is followed, as is done in this work. The thrusters are controlled by 8 electronic solenoid valves controlled by a Raspberry Pi 3 onboard computer. The reaction wheel is also controlled by the onboard computer via an Ethernet connection. Similar to *Elissa* in Sub-Section 3.1, the control during experiments is done using the hardware-in-loop method where a lab computer performs all the control computations and sends commands to the onboard computer via WiFi with *ROS2 Humble* as the communication middleware. The onboard computer performs the

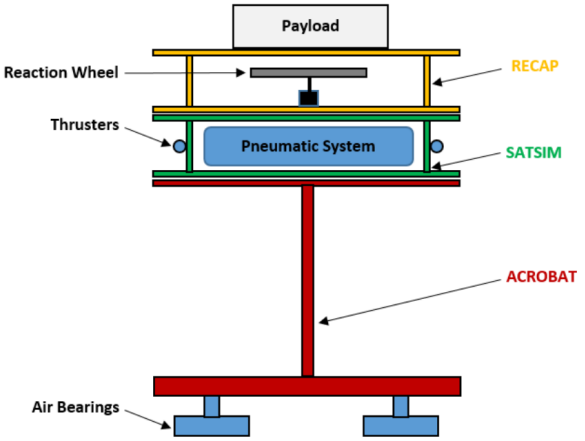


Figure 3: REACSA Schematic. From [3]

commanded actuation commands and provides feedback. Full-state feedback of the system is obtained using the *Vicon* motion capture system. The closed-loop control is run at $\approx 100\,\text{Hz}$.

The REACSA system is modeled as a double rigid body (main body and reaction wheel) with the following state vector $\mathbf{x} \in \mathbb{R}^7$:

$$\mathbf{x} = [\mathbf{q} \quad \dot{\mathbf{q}}]^T = [x \quad y \quad \theta_z \quad \dot{x} \quad \dot{y} \quad \omega_z \quad \omega_{RW}]^T \tag{10}$$

Here, the extra term ω_{RW} is the reaction wheel velocity while the rest are similar to that presented earlier in Sub-Section 3.1. The control vector $\mathbf{u}$ is then defined as:

$$\mathbf{u} = [\tau \quad f_0 \quad f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5 \quad f_6 \quad f_7]^T \tag{11}$$

where τ is the torque applied to the reaction wheel while f_n is the force applied at the n^{th} thruster. Now, we can write the dynamics of the system as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{0}^{3\times3} & \mathbf{I}^{3\times3} & 0 \\ & \mathbf{0}^{4\times7} & \end{bmatrix} \mathbf{x} + \begin{bmatrix} & & & & \mathbf{0}^{3\times7} & & & & \\ 0 & \frac{-s_\theta}{m} & \frac{s_\theta}{m} & \frac{-c_\theta}{m} & \frac{c_\theta}{m} & \frac{s_\theta}{m} & \frac{-s_\theta}{m} & \frac{c_\theta}{m} & \frac{-c_\theta}{m} \\ 0 & \frac{c_\theta}{m} & \frac{-c_\theta}{m} & \frac{-s_\theta}{m} & \frac{s_\theta}{m} & \frac{-c_\theta}{m} & \frac{c_\theta}{m} & \frac{s_\theta}{m} & \frac{-s_\theta}{m} \\ \frac{-1}{I_b} & \frac{r}{I_b} & \frac{-r}{I_b} & \frac{r}{I_b} & \frac{-r}{I_b} & \frac{r}{I_b} & \frac{-r}{I_b} & \frac{r}{I_b} & \frac{-r}{I_b} \\ \frac{1}{I_w} & & & & \mathbf{0}^{1\times6} & & & & \end{bmatrix} \mathbf{u} \tag{12}$$

here, m is the total system mass, I_w and I_b are the moment of inertia of the reaction wheel and the overall system respectively, and s_θ and c_θ denote the sine and cosine of the respective angles.

We can now use the trajectory optimization and control methods given in Section 2 for this system as well. For the trajectory optimization formulation, it is assumed that the thrusters can provide continuous force instead of binary on/off actuation. This is later converted to binary commands using a Sigma-Delta modulator [30]. The trajectory optimization problem is written as:

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{U}} \left\{ \sum_{k=1}^{N} \mathbf{u}_k \mathbf{R} \mathbf{u}_k^T \right\} \quad \forall k \in [1, N-1] \text{ s.t.} \\ &\mathbf{x}(0) = \mathbf{x}_{init}, \quad \mathbf{x}(t_f) = \mathbf{x}_{final} \\ &\mathbf{x}_{min} \leq \mathbf{x}_k \leq \mathbf{x}_{max}, \quad \mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \\ &\mathbf{x}_{k+1} - \mathbf{x}_k = \frac{\Delta t}{6}(\mathbf{f}_k + 4\mathbf{f}_{k+1/2} + \mathbf{f}_{k+1}) \end{aligned} \tag{13}$$

where

$$\begin{aligned} \mathbf{x}_{k+1/2} &= \frac{1}{2}(\mathbf{x}_k + \mathbf{x}_{k+1}) + \frac{\Delta t}{8}(\mathbf{f}_k - \mathbf{f}_{k+1}) \\ \mathbf{u}_{k+1/2} &= \frac{1}{2}(\mathbf{u}_k + \mathbf{u}_{k+1}) \end{aligned}$$

In contrast to the backward-Euler integration scheme used in Sub-Section 3.1, Hermite-Simpson is chosen as the transcription method. This non-linear program is then solved using the *IPOPT* solver [27]. For control, a TVLQR is used whose output is routed via a Sigma-Delta modulator as mentioned before. For state estimation, a Kalman Filter is used. The control architecture used for the experiments can be seen in Figure 4.

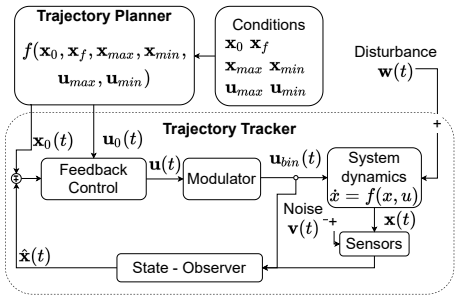


Figure 4: REACSA Control Architecture. From [2]

3.3. REACSA with Robot Arm

As an upgrade to the REACSA system, a robot arm was also attached to the system along with another passive floating system: MANTIS. This setup can then be used to study and test free-floating/free-flying multibody dynamics which is required for ADR/OOS missions. The system setup can be seen in Figure 5. The robot arm is a custom 3-DoF planar robot arm with 3 Quasi-Direct Drive (QDD) actuators (T-Motor AK80-6) which provide direct torque control and feedback along with low friction and backdriveability. These are connected with three $0.4\,\text{m}$ links which are made from carbon fiber plates with a porous substrate in between for extremely lightweight ($100\,\text{g}$) and rigid links. The robot arm is controlled via

a CAN network with closed-loop control speed capable at 1 kHz using the driver from [25]. The CAN bus was attached to the on-board computer Raspberry Pi using a USB-CAN adaptor and the closed-loop control frequency during experiments was kept at 100 Hz. For the control architecture, the previous architecture from Figure 4 was expanded to include command and feedback from the robot arm using a robot arm driver. The control frequency was kept the same. The system model and the controller implemented can be seen in more detail in [1, 26] which is in line with Section 2. During the experimental campaign, multiple such disturbances were added to the system at various locations of the system. The LQR controller gains were tuned such that the velocity gains were significantly higher than the position gains. This was done to emulate the use of the LQR controller for a post-capture detumble scenario. Thus, due to these gains, the LQR controller was able to reduce the angular velocities of the system to below the testbed noise levels and hence *detumble* successfully.

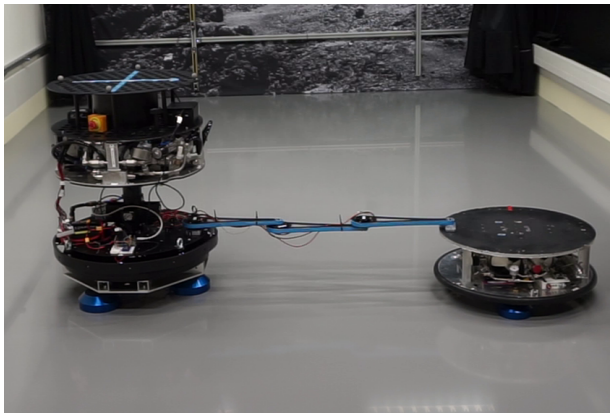


Figure 5: Free-Floating Multibody Robot Setup at ESA-ESTEC. On the left is the REACSA platform and on the right is the MANTIS platform. They are connected with a custom 3-DoF planar robot arm.

4. EXPERIMENTAL RESULTS

4.1. Elissa

The state and actuation cost weight matrices were determined and tuned experimentally by trial and error. The following matrices were used:

$$\mathbf{Q} = \begin{bmatrix} 50 & 0 & 0 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.001 \end{bmatrix} \tag{14}$$

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10 \end{bmatrix} \tag{15}$$

With the following cost matrices, the *Elissa* free-flyer was able to track an almost circular trajectory while keeping one side facing the center of the circle to simulate an observation mission maneuver. The snapshots of the trajectory along with the free-flyer in various positions can be seen in Figure 6. The tracking performance of the controller with respect to the optimal trajectory computed can be seen in Figure 7. While the position tracking is good, the velocity tracking is observed to have slightly curtailed performance with this controller and the experimentally tuned gains. We estimate this can be improved in the future with better system identification methods.

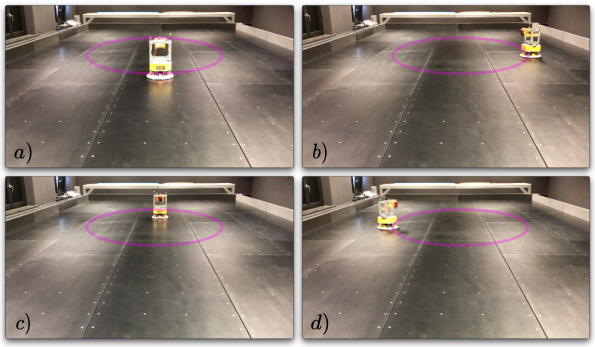
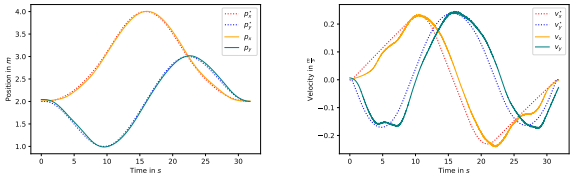


Figure 6: Snapshots of the Free-flyer moving along the nominal trajectory



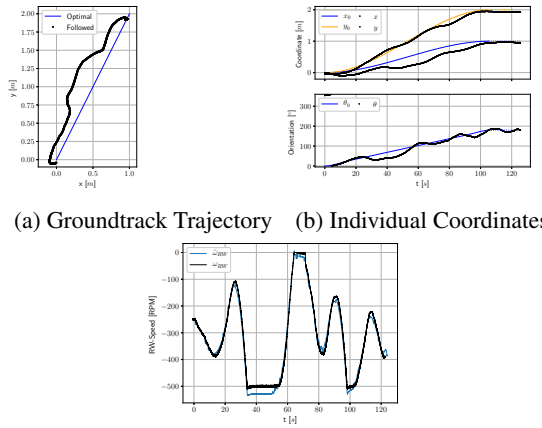
(a) Optimal vs. measured position.

(b) Optimal vs. measured linear velocity.

Figure 7: Elissa TVLQR nominal stabilization results

4.2. REACSA

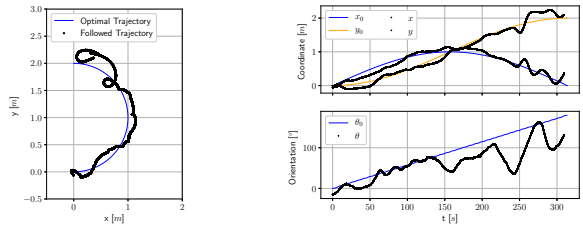
Similar to Sub-Section 4.1, the state and cost matrices for the experiments with REACSA were also estimated by trial and error during experiments. Two trajectories were devised for the experiments: a straight line of 2.2 m with 180° turn and a semi-circle with one side pointing at the center of the circle to emulate an observation mission. Furthermore, as the reaction wheel experiences stiction at low speeds, it is spun up to half its rated velocity before the experiment begins. The results of these experiments can be seen in Figure 8 and Figure 9 respectively.



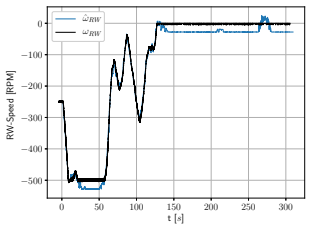
(a) Groundtrack Trajectory (b) Individual Coordinates

(c) Reaction wheel Speed

Figure 8: Experimental Results from the Straight Line Experiment with $180°$ turn. From [2].



(a) Groundtrack Trajectory (b) Individual Coordinates



(c) Reaction wheel Speed

Figure 9: Experimental Results from the Semi-Circle Experiment. From [2].

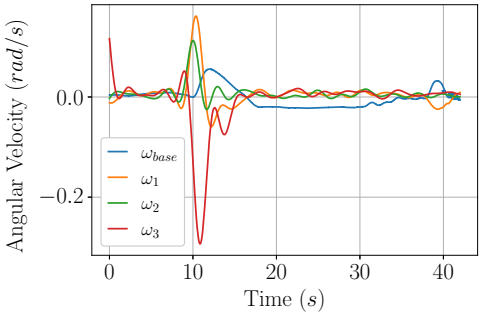
The system was able to track the given trajectories and reject the disturbances from the external environment (such as the slightly uneven floor [2, 4]). However, the performance on the real system was degraded when compared to earlier simulations: average linear errors of $0.325\,\mathrm{m}$ and $0.315\,\mathrm{m}$ and average angular errors of $23.8°$ and $51.2°$ for the linear and semi-circle trajectory respectively. Furthermore, it can be seen from Figure 8c and Figure 9c that the reaction wheel is saturated multiple times during the trajectory execution and control. After reaction wheel saturation, the attitude control is provided by thrusters which degrades the attitude control performance significantly. It can be concluded that the reaction wheel inertia is smaller than what would be desired for good control authority for the REACSA system given the external disturbances caused by the unevenness of the floor.

4.3. REACSA with Robot Arm

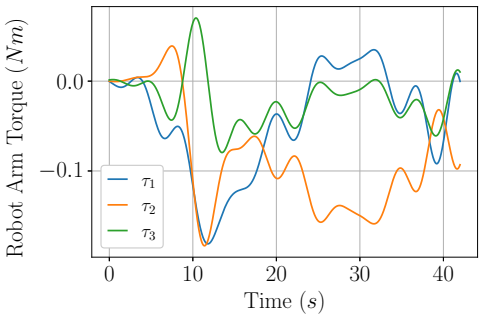
Similar to earlier experimental results, the cost matrices were also determined using trial and error during experiments. For the experimental results shown in Figure 10, the system was set up in zero configuration as shown in Figure 5, and a disturbance was added to the system in the form of pushing it with a stick at the center of the robot arm. This disturbance can be seen in Figure 10 at $\approx 11\,\mathrm{s}$. The LQR controller recovers from this large disturbance and the system returns to its initial velocities which are non-zero due to the disturbance from the floor. This can be considered as the base-level noise of the system during experiments.

5. CONCLUSION

This paper shows experimental validation of the LQR controller for different free-floating systems. The novelty of this work lies in demonstrating the robustness and flexibility of the LQR controller on various platforms that have different actuation modes: continuous (propeller, robot arm, reaction wheel) and binary (thrusters), or single or multibody systems, or mixed or non-mixed actuation modalities. While experimental validations are not as common in literature, a similar methodology applied to 3 different testbeds demonstrates the versatility of the trajectory optimization and stabilization methods given here. The next steps to further this research are to experimentally demonstrate and validate the Region of Attraction (RoA) of the LQR controllers using the Lyapunov stability criteria, including bilateral contact dynamics and mapping to the controller synthesis and RoA analysis using the jump Riccati equation, extending the LQR formalization to Mixed-Integer Model Predictive Control, and furthermore testing on even more platforms such as underwater platforms by including the hydrodynamic forces in the model to execute *hydrobatic* maneuvers.



(a) Base and Robot Arm joint angular velocities during a single-push experiment



(b) Robot Arm torques during a single-push Experiment

Figure 10: LQR Stabilization of the Multibody System during a single-push Experiment. The push disturbance occurred at $\approx 11\,\mathrm{s}$

ACKNOWLEDGMENTS

The first and second authors would like to acknowledge the support of M-RoCk (Grant No.: FKZ 01IW21002) and AAPLE (Grant Number: 50WK2275) and the first author would also like to acknowledge the funding from Cooperants (Grant Number: 68GX21003H) and the support from the European Union's Horizon 2020 research and innovation programme 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] A. Bredenbeck, S. Vyas, W. Suter, M. Zwick, D. Borrmann, M. Olivares-Mendez, and A. Nüchter. Finding and following optimal trajectories for an overactuated floating robotic platform. In *Proceedings of ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA)*, Noordwijk, Netherlands, 2022.

[3] Anton Bredenbeck. *SatSim-REACSA - A Floating Platform for Simulation of Micro-Gravity With Three Degrees of Freedom*. Master Thesis, Julius-Maximilians University Würzburg, Würzburg, February 2022.

[4] Anton Bredenbeck, Shubham Vyas, Martin Zwick, Dorit Borrmann, Miguel Olivares-Mendez, and Andreas Nüchter. Trajectory optimization and following for a three degrees of freedom overactuated floating platform. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4084–4091. IEEE, 2022.

[5] Angel Flores-Abad, Lin Zhang, Zheng Wei, and Ou Ma. Optimal capture of a tumbling object in orbit using a space manipulator. *Journal of Intelligent & Robotic Systems*, 86:199–211, 2017.

[6] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

[7] J Graham, R Ravindran, and K Knapp. Space manipulators-present capability and future potential¡ 149¿ space shuttle remote handling system. In *Conference on Advanced Technology for Future Space Systems*, page 903, 1979.

[8] Francis James, Shubham Vyas, Puneeth Bandikatla, and P Mithun. Design and development of an earth based experimental setup for testing algorithms on space robots. In *Proceedings of the 2015 Conference on Advances in Robotics*, pages 1–6, 2015.

[9] Francis James, Shubham Vyas, Puneeth Bandikatla, P Mithun, and S.V Shah. Design and development

of an earth based experimental setup for testing algorithms on space robots. In *Proceedings of the 2015 Conference on Advances in Robotics*, pages 1–6, 2015.

[10] Francis James, Suril V Shah, Arun K Singh, K Madhava Krishna, and Arun K Misra. Reactionless maneuvering of a space robot in precapture phase. *Journal of Guidance, Control, and Dynamics*, 39 (10):2419–2425, 2016.

[11] J-C Liou. An active debris removal parametric study for leo environment remediation. *Advances in space research*, 47(11):1865–1876, 2011.

[12] Lasse Maywald. *Design and Qualification of Hardware-in-the-Loop Testbed to Experimentally Evaluate Spacecraft Proximity Operations*. Study Thesis, TU Braunschweig, Braunschweig, August 2020.

[13] AI Nazarenko and IV Usovik. Space debris in low earth orbits region: Formation and reduction process analysis in past decade. *Acta Astronautica*, 194:383–389, 2022.

[14] Thai Chau Nguyen Huynh. *Adaptive reactionless control of a space manipulator for post-capture of an uncooperative tumbling target*. PhD thesis, McGill University, 2013.

[15] Evangelos Papadopoulos and Steven Dubowsky. On the nature of control algorithms for free-floating space manipulators. *IEEE Transactions on Robotics and Automation*, 7(6):750–758, 1991.

[16] Evangelos Papadopoulos, Iosif S Paraskevas, Thaleia Flessa, Kostas Nanos, Georgios Rekleitis, and Ioannis Kontolatis. The ntua space robot simulator: Design & results. In *ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA 2008)*, 2008.

[17] Tomasz Rybus, T Barciński, J Lisowski, J Nicolau-Kukliński, K Seweryn, M Ciesielska, K Grassmann, J Grygorczuk, M Karczewski, M Kowalski, et al. New planar air-bearing microgravity simulator for verification of space robotics numerical simulations and control algorithms. In *12th ESA Symposium on Advanced Space Technologies in Robotics and Automation*, 2013.

[18] Tomasz Rybus, Mateusz Wojtunik, and Fatina Liliana Basmadji. Optimal collision-free path planning of a free-floating space robot using spline-based trajectories. *Acta Astronautica*, 190:395–408, 2022.

[19] Jana L Schwartz, Mason A Peck, and Christopher D Hall. Historical review of air-bearing spacecraft simulators. *Journal of Guidance, Control, and Dynamics*, 26(4):513–522, 2003.

[20] Suril V Shah, Inna Sharf, and Arun Misra. Reactionless path planning strategies for capture of tumbling objects in space using a dual-arm robotic system. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, page 4521, 2013.

[21] Russ Tedrake. Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832). Downloaded on 2021-06-01 from http://underactuated.mit.edu/.

[22] C. Trentlage, J. Yang, M. K. Ben Larbi, C. A. de Alba Padilla, and E. Stoll. The elissa laboratory: Free-floating satellites for space-related research. In *Deutscher Luft- und Raumfahrtkongress*, Friedrichshafen, Germany, 2018.

[23] Yoji Umetani, Kazuya Yoshida, et al. Resolved motion rate control of space manipulators with generalized jacobian matrix. *IEEE Transactions on robotics and automation*, 5(3):303–314, 1989.

[24] Josep Virgili-Llop, Costantinos Zagaris, Richard Zappulla, Andrew Bradstreet, and Marcello Romano. A convex-programming-based guidance algorithm to capture a tumbling object on orbit using a spacecraft equipped with a robotic manipulator. *The International journal of robotics research*, 38 (1):40–72, 2019.

[25] S Vyas. C++ Motor Driver for Mini Cheetah-type Actuators from T-Motor/CubeMars. `https://github.com/dfki-ric-underactuated-lab/mini-cheetah-tmotor-can`, 2023. Accessed: 2023-19-09.

[26] Shubham Vyas, Lasse Maywald, Shivesh Kumar, Marko Jankovic, Andreas Mueller, and Frank Kirchner. Post-capture detumble trajectory stabilization for robotic active debris removal. *Advances in Space Research*, 72(7):2845–2859, 2023.

[27] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

[28] Markus Wilde, Stephen Kwok Choon, Alessio Grompone, and Marcello Romano. Equations of motion of free-floating spacecraft-manipulator systems: an engineer's tutorial. *Frontiers in Robotics and AI*, 5:41, 2018.

[29] Juntang Yang, Yogesh Khedar, Mohamed Khalil Ben-Larbi, Jan Backhaus, Astrid Lampert, Ulf Bestmann, and Enrico Stoll. Concept and feasibility evaluation of distributed sensor-based measurement systems using formation flying multicopters. *Atmosphere*, 12(7):874, 2021.

[30] Richard Zappulla, Josep Virgili-Llop, Costantinos Zagaris, Hyeongjun Park, and Marcello Romano. Dynamic Air-Bearing Hardware-in-the-Loop Testbed to Experimentally Evaluate Autonomous Spacecraft Proximity Maneuvers. *Journal of Spacecraft and Rockets*, 54(4):825–839, July 2017. ISSN 0022-4650, 1533-6794. doi: 10.2514/1.A33769.

[31] Martin Zwick, Irene Huertas, Levin Gerdes, and Guillermo Ortega. ORGL – ESA'S Test Facility For Approach and Contact Operations in Orbital and Planetary Environments. In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, volume 6, Madrid, Spain, June 2018.