Design of an attitude control system enabled by the development of a digital twin of a CubeSat equipped with a robotic manipulator

Bartłomiej Koszek, Álvaro Román Sánchez
KU Leuven University
bartlomiej.koszek@student.kuleuven.be, alvaro.romansanchez@student.kuleuven.be

ABSTRACT

In this research, the authors create a digital twin of a CubeSat equipped with a robotic manipulator to study in-orbit servicing and active debris removal. This study focuses on small robotic platforms where the inertia of the manipulator and the spacecraft body are comparable. Due to the size reduction and therefore inertia reduction, these platforms face large and often unknown dynamic coupling forces between the robotic manipulator and the spacecraft body. Control laws for an example vehicle are explored based on realistic scenarios. The final improvements to the attitude control system include a method of re-tuning the control gains based on the robot's pose and a feedforward controller to counteract the disturbance that results from the motion of the robot arm. Furthermore, the actuator effort and overall momentum requirement for this advanced controller have been evaluated, which can be used to inform component selection.

Keywords: Attitude Control, Rendezvous Proximity Operations, Space Robotics, Controller Tuning, Recursive Newton-Euler Algorithm

INTRODUCTION

Achieving space mission objectives is highly reliant on attitude control. Imaging payloads, communication, etc. require fine control over the orientation of a spacecraft: its attitude. The motion between two attitude setpoints is important, just as the steady state error when pointing in one direction. Depending on the mission types, the requirements for the overall pointing error can be different. The selection of control methodology is based on these requirements and spacecraft specific dynamics.

Dynamics of the spacecraft refer to the inertia parameters of the spacecraft and its weight. In the space environment, these aspects impact the way a spacecraft spins around its axis and also whether the system behaves in a linear or nonlinear way. ADCS systems are built up using actuators that can impart torques on the spacecraft's axis (XYZ), and the amount of push each torque corresponds to is encoded in the dynamics. Additionally, a sensor is used to measure the orientation and angular velocity. Sun sensors, horizon sensors, and star trackers are used to measure the attitude, and gyroscopes can be used to measure the angular velocity. A control law is then implemented to minimize the difference between the desired attitude and the measured attitude. Many spacecraft often have several control laws implemented and select between them depending on the scenario at play.

For most missions today, the satellite dynamics remain constant during the execution of mission objectives, and therefore, the control laws are developed with a focus on rejecting external torques such as solar radiation pressure, magnetic gradients, gravity gradient, and aerodynamic torques [1]. Next generation missions such as rendezvous and proximity operations (RPO) pose a challenge for these traditional control methodologies, namely as these spacecraft buses are more complicated with robotic manipulators, thrusters, and docking elements that may contribute to changing these satellite dynamics while executing the given mission.

Platforms most affected by these changing dynamics are small satellites equipped with a robotic arm for in-space manipulation tasks. When the robotic manipulator mass makes up a significant portion of the total system mass, any motion of the manipulator changes both the spacecraft dynamics and applies a disturbance torque on the spacecraft, which affects the overall pointing error. Within our study, we aim to study the impacts of this coupling between the manipulator and satellite and how it affects the attitude controller design.

BACKGROUND

A spacecraft's attitude can be parametrized using Euler angles (ϕ, θ, ψ). Each angle represents a rotation about one of the primary axes of the spacecraft's body frame. Figure 1 shows each axis of the spacecraft (x', y', z'). Another alternative way of parameterizing the attitude is the equivalent axis representation, which utilizes a reference rotation axis and an angle α. This is shown in Figure 2.

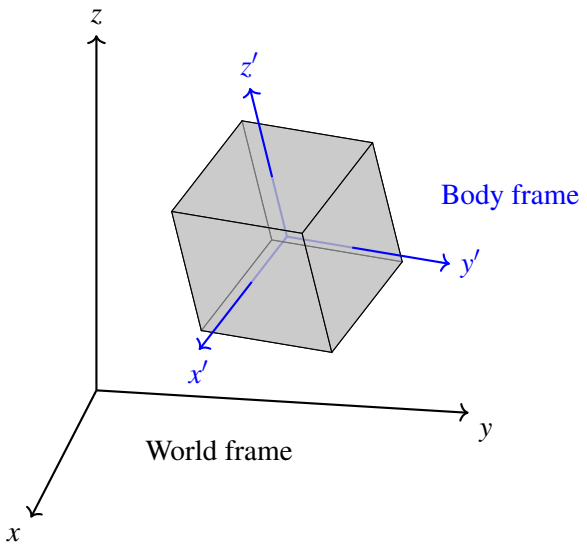

Figure 1. 3D representation of a body frame (x', y', z') with respect to a world frame (x, y, z).

Within the space environment, all rotations happen around the center of mass, which is the origin of the spacecraft body frame. A satellite's attitude parameters are coupled to each other based on the inertia parameters. Equation 1 highlights the dynamic response of angular velocity around each axis caused by an applied torque [2].

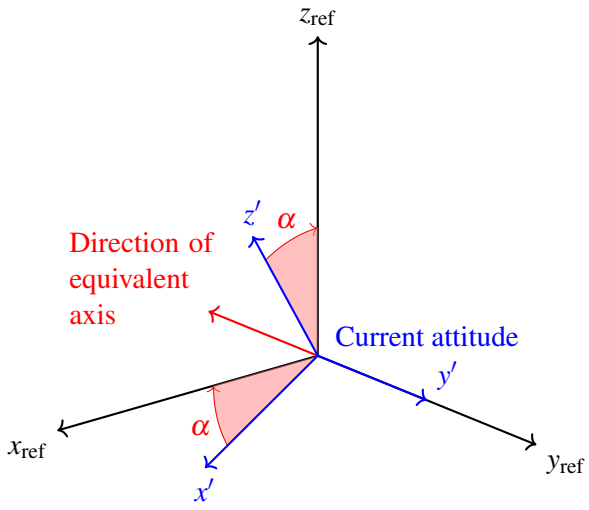

Figure 2. Illustration of the equivalent axis and α.

$$\begin{aligned} \tau'_x &= I_{xx}\dot{\omega}'_x + (I_{zz} - I_{yy})\,\omega'_y\omega'_z \\ \tau'_y &= I_{yy}\dot{\omega}'_y + (I_{xx} - I_{zz})\,\omega'_z\omega'_x \\ \tau'_z &= I_{zz}\dot{\omega}'_z + (I_{yy} - I_{xx})\,\omega'_x\omega'_y \end{aligned} \tag{1}$$

Where I_{xx}, I_{yy}, I_{zz} refer to the principal moments of inertia of the spacecraft, and $\omega'_x, \omega'_y, \omega'_z$ refer to angular velocities expressed in the body frame. From Equation 1 it can be seen that there is a coupling between each axis of rotation of the spacecraft, and the magnitude of this coupling is dictated by the relations between the inertia parameters of the spacecraft.

For modern spacecraft, the inertia parameters and center of mass (collectively referred to as the system's dynamics) generally remain constant over time. In most missions, the primary source of variation is typically the deployment of solar panels. Once deployed, most solar panel systems remain static, and the spacecraft's dynamics stabilize, remaining unchanged post-deployment. Furthermore, in cases where solar panels are externally mounted to the satellite bus, no significant change occurs in the spacecraft's dynamics.

Another factor that could alter the spacecraft's dynamics is the presence of thrusters and their fuel. Depending on the fuel used, there may be a fluid within the tank that can move during operation. This may cause the mass distribution of the satellite to change over time, varying the dynamics. To better predict how dynamics evolve, sloshing models are often used. These models, based on computational fluid dynamics, attempt to account for fuel movements within the tank, though accurately modeling fluid behavior is highly complex and prone to variability [3]. As a result, such dynamic influences are typically addressed with a suitable control system. Additionally, since the fuel tank is located at a fixed point on the spacecraft, simplifications can be made, such as assuming a uniform fuel distribution and using the average weight of the fuel. These assumptions can be updated as the fuel depletes over time. However, the impact of fuel sloshing on the spacecraft's overall dynamics remains limited, as the dynamics only shift within a range defined by the spacecraft's full and empty fuel states. This reduces the significance of this problem within the ADCS design workflow as a sufficiently robust controller can be selected to counteract these dynamic changes and maintain system performance. Such a controller can fulfill all relevant mission requirements for the entire mission lifespan, even as the fuel depletes over time.

Satellites designed for RPO have inherently more complex dynamics. The inclusion of robotic manipulators, docking elements, and thrusters leads to dynamic variations that change while carrying out mission objectives. The most significant variation comes from the robotic manipulator, as it typically moves the most mass within the spacecraft's reference frame. This causes substantial changes in both spacecraft inertia and center of mass (COM). As the robot pose is altered, the position of the COM changes, which also means that the center of rotation of the spacecraft changes. Inertia is defined by the product of mass and distance from the center of rotation. During manipulation, the mass distribution changes within the spacecraft body frame as well as the location of the origin of the frame.

The second major source of dynamic variation arises from interactions with a resident space object (RSO), which can be either cooperative (actively attempting to dock) or non-cooperative (where many parameters of the RSO are unknown prior to contact). Once a spacecraft captures an RSO, the mass and geometry properties of the RSO change the spacecraft dynamics. Additionally, the angular momentum stored in the RSO is transferred to the spacecraft, which results in disturbance torques. Depending on how the capture operation is conducted, the magnitude of these disturbance torques can vary. If an uncooperative RSO were to be spinning rapidly and then it were to be captured by the spacecraft, a large transfer of angular momentum would occur [4]. Ultimately, this would result in increasing the pointing error of the spacecraft and could potentially result in a loss of control over the spacecraft should the magnitude of the angular momentum transfer be significantly high.

PRIOR WORK

Space Vehicles capable of executing RPO missions have been theorized, and some even put into service; however, most of these have been on manned missions [5]. Several studies into control strategies for autonomous RPO systems have been performed in the past. The main challenge of these studies is modeling of the dynamics of spacecraft systems with changing inertia properties. Some studies have explored modeling and control of large flexible spacecraft, where a disturbance observer has been developed to estimate the external disturbance that results from the flexible parts of the spacecraft. This observer is then used to inform the control policy of the spacecraft [6]. Additionally, [7] focused more on how control policies can utilize a similar disturbance observer.

Other studies went more in-depth on modeling and overall system analysis for large flexible spacecraft, such as [8]. In this study, an analytical method was developed, which was later used to inform the control law synthesis based on mission requirements. Several studies also discuss the control techniques in relation

to combining several aspects of such missions, such as [9] where a camera was simulated along with the vehicle, [7] , which focused on integrating ADCS with sensors for close proximity, and [10] , which focused on combining arm trajectory planning with simulated camera feedback.

Several projects focused on developing the hardware for such a mission have also been developed. A vehicle developed in a study [11] focused on the manipulation aspect of RPO missions, however, this study did not include attitude control within the scope of the project. Other vehicles have been proposed, such as [12] and [13] , which combine robotic manipulators with satellites.

RESEARCH OBJECTIVES

The objective of this study was to utilize a digital twin of a theoretical RPO vehicle. A theoretical vehicle has been chosen as a realistic case study. The vehicle's total mass was 302 kg, with the manipulator weighing 102 kg. The manipulator's geometry selected for this study was based on a 7 DOF ABB IRB 14050 robot arm. The use of a realistic robotic arm enables future studies to be conducted with hardware in the loop. The spacecraft's body has been chosen as a cube of side length 700 mm. Reaction wheels have been added, one for each of the primary axes, to impart control torques on the system. The CAD data for this vehicle includes solar panels; however, for the purposes of this study, they are only visually rendered and are not included in the physical properties. The full Vehicle can be seen in Figure 3.

The scope was limited to only focusing on counteracting the disturbances coming from the motion of the arm and RSO capture. External disturbance torques such as atmospheric drag, magnetic gradient, and solar radiation pressure were not included in this study, however, the simulator can easily be extended to include these. Additionally, thrusters have not been included in the scope, however, the simulator does account for translational motion. Disturbances which result from interactions within the simulator may result in motion, like in the case of RSO capture, however,r only control over the vehicle's attitude is explored.

The authors have developed a digital twin simulation combining control aspects and multi-body simulation by utilizing a co-simulation approach between MATLAB Simulink and Siemens NX Mechatronics Concept Designer (MCD). Vehicle dynamics and the robot arm were modeled within the engineering software MCD, which enabled fast iteration and working directly with CAD data. Actuators, sensors, and rigid bodies are created within MCD, and several signals are then connected to the controller in Simulink. Within Simulink, several algorithms have been developed to produce control outputs that are sent back to MCD, where the control actions are executed on the system. Additional details of this simulation environment are further described in [14].

The objective of this research was to investigate control techniques that could be implemented for the attitude control of an RPO mission to counteract the disturbance caused by the motion of the manipulator and to ensure the performance of the ADCS with varying inertial properties of the spacecraft. Within this study, we also investigated how RSO capture influences the performance of the ADCS. For the purpose of controller design, the feedback from two sensors has been taken into account: an attitude sensor (a star-tracker) and an angular velocity sensor (a gyroscope). Within the digital twin, the star tracker was approximated as providing an ideal attitude, albeit at a reduced frequency with a zero-order hold, 10 Hz, based on the state-of-the-art commercially available star trackers [15]. Due to the availability of very high update rate gyroscope sensors, within the digital twin, this sensor was modeled as a continuous signal with added noise based on a commercially available unit. The noise characterization used in this research was based on [16].

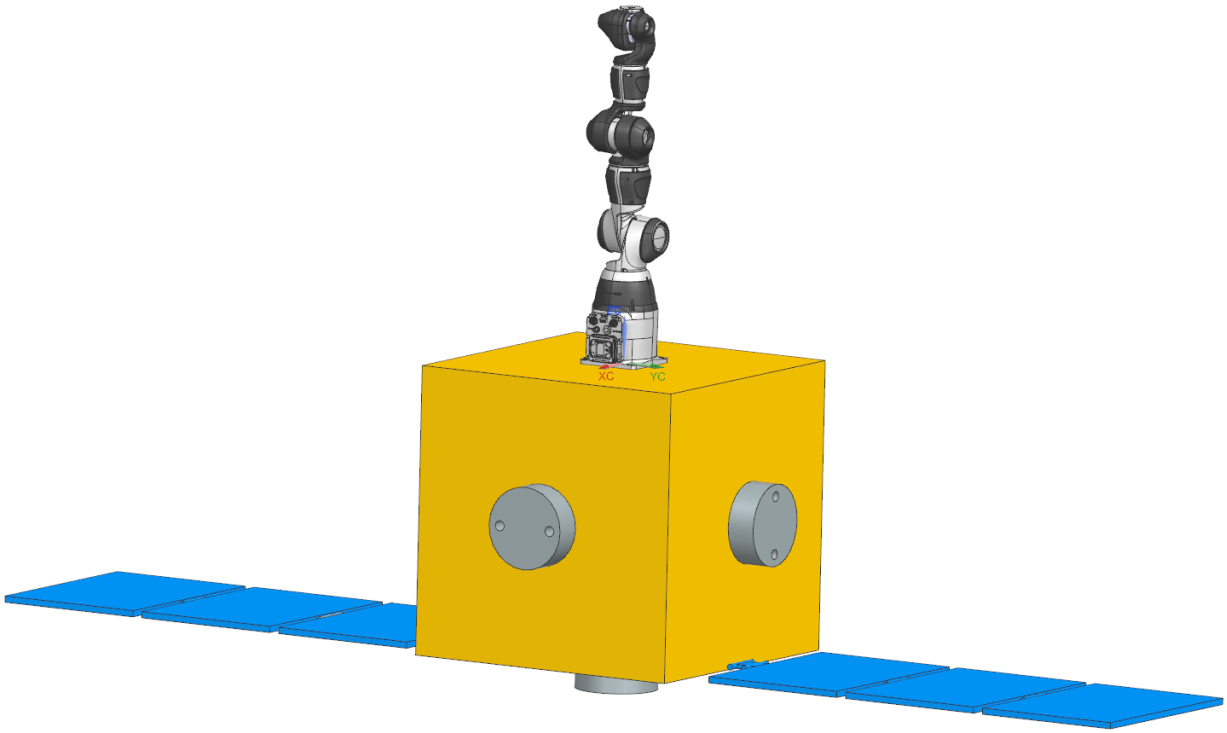

Figure 3. Theoretical RPO vehicle

BASIC CONTROL SYSTEM

The initial approach towards the design of the control system was to implement a simple feedback P controller, where the attitude error was calculated using the Euler angles. This was quickly found to be inappropriate given the existence of representational singularities in the Euler angles, causing the system to be unstable when the satellite's orientation was close to such singularities. Therefore, quaternions were chosen to represent attitude due to their lack of singularities. A quaternion-based closed-loop control system largely based on [17] was implemented, with an added step of changing the reference frame of the control input torque. The reason for this is that the torque calculated by the P controller is expressed in the world frame, while the torque given by the D controller is expressed in the body frame since the gyroscope provides angular velocity measurements in the body frame only. Conversion of total control input torque to the body frame is also convenient, as it is easier to calculate reaction wheel speed inputs this way.

Controller tuning

To find suitable control gains, a model of the plant dynamics is needed, which has been linearized to find its state-space representation in the form $\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t)$. The model discussed in [17] uses the Euler-Newton equations to express the rotational dynamics of a rigid body (equation (3)), while quaternion kinematics are used to relate a rigid body's angular velocity and attitude quaternion to the time derivative of the attitude quaternion (equation (2)). This model is well-suited, as it omits the translational dynamics of the rigid body, which are outside the scope of this research due to the lack of thrusters.

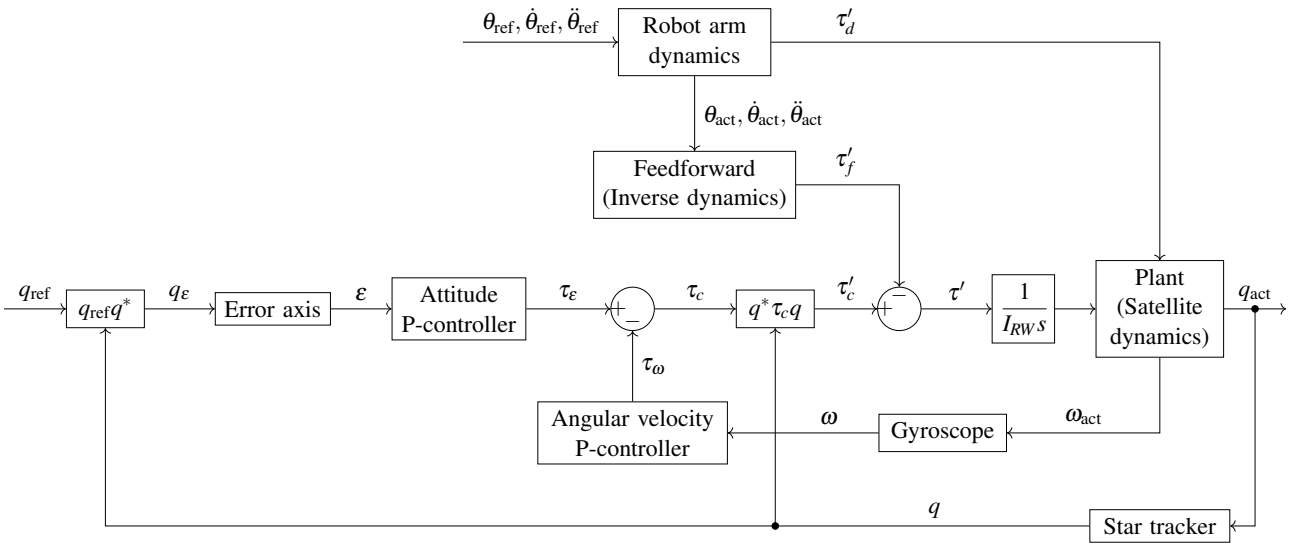

Figure 4. Block diagram of the full control system workflow.

$$\dot{q} = \frac{1}{2} q \begin{bmatrix} 0 \\ \omega' \end{bmatrix} \tag{2}$$

$$\dot{\omega}' = I^{-1}\left(\tau - \omega' \times I\omega'\right) \tag{3}$$

The above expressions provide a set of 7 equations of motion, the first of which is redundant due to the nature of quaternions. To make the system controllable, this equation is, from now on excluded from the model, which makes the rank of the controllability matrix equal to the number of states, namely 6. We can therefore define the state vector $x(t) = \begin{bmatrix} q_1 \; q_2 \; q_3 \; \omega'_x \; \omega'_y \; \omega'_z \end{bmatrix}^T$ and the input vector $u(t) = \begin{bmatrix} \tau'_x \; \tau'_y \; \tau'_z \end{bmatrix}^T$. The matrices **A** and **B** are then:

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{I_{yy}-I_{zz}}{I_{xx}}\omega'_z & \frac{I_{yy}-I_{zz}}{I_{xx}}\omega'_y & 0 & 0 & 0 \\ -\frac{I_{xx}-I_{zz}}{I_{yy}}\omega'_z & 0 & -\frac{I_{xx}-I_{zz}}{I_{yy}}\omega'_x & 0 & 0 & 0 \\ \frac{I_{xx}-I_{yy}}{I_{zz}}\omega'_y & \frac{I_{xx}-I_{yy}}{I_{zz}}\omega'_x & 0 & 0 & 0 & 0 \\ \frac{q_0}{2} & -\frac{q_3}{2} & \frac{q_2}{2} & 0 & \frac{\omega'_z}{2} & -\frac{\omega'_y}{2} \\ \frac{q_3}{2} & \frac{q_0}{2} & -\frac{q_1}{2} & -\frac{\omega'_z}{2} & 0 & \frac{\omega'_x}{2} \\ -\frac{q_2}{2} & \frac{q_1}{2} & \frac{q_0}{2} & \frac{\omega'_y}{2} & -\frac{\omega'_x}{2} & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

A and **B** must then be evaluated at a chosen linearization point, which in this case was set to be the set-point attitude with an angular velocity of zero. A full state feedback controller **K** is then selected to be used in the feedback control law $u(t) = -\mathbf{K}x(t)$. In this research the authors used pole-placement to find **K**, however a Linear-Quadratic Regulator (LQR) may alternatively be used if the necessary cost matrices are defined. This completes the tuning process.

For the implementation of the aforementioned control law in the proposed control system, the 3×6 controller $\mathbf{K} = [\mathbf{P}\,\mathbf{D}]$ is split into its 3×3 sub-matrices **P** and **D**, which contain the control gains pertaining to the attitude and the angular velocity, respectively. Finally, as shown in Figure 4, both controllers perform a matrix multiplication to obtain their corresponding output torque:

$$\begin{aligned} \tau_\varepsilon &= \mathbf{P}\varepsilon \\ \tau_\omega &= \mathbf{D}\omega \end{aligned} \tag{4}$$

Problems with the basic controller

Two main problems exist with the controller described above. The first one is that the controller is purely reactive, therefore it can only counteract attitude errors once they already exist. This makes it suitable to track changes in the attitude setpoint which typically occur very slowly (for instance on nadir pointing satellites), but not suitable to counteract disturbance torques caused by movements of the robot arm, which may occur much faster.

The second problem is that the controller gains found only provide the desired response for a single pose of the robot arm and a single end-effector load. If a different configuration is used, the control gains may no longer be representative of the desired pole locations, and the behavior may deviate significantly from what is expected. This causes a drop in overall performance.

FEEDFORWARD

To solve the first of the problems identified above, a feedforward approach is taken. It consists of predicting the disturbance torques τ'_d caused by the robot arm on the satellite body while (or before) they are applied, then applying equal but opposite control input torques τ'_f. This requires a method for analytically calculating such torques and a very good understanding of the robot arm's dynamics.

As shown in Figure 4, the joints of the robot arm are provided with a reference trajectory $\theta_{\text{ref}}, \dot{\theta}_{\text{ref}}, \ddot{\theta}_{\text{ref}}$, which is then fed into the simulation environment where the actual (true) trajectory $\theta_{\text{act}}, \dot{\theta}_{\text{act}}, \ddot{\theta}_{\text{act}}$ is measured using the joints' encoders. Given the true trajectory, the following well-known expression provides the arm's motor torques τ_m:

$$\tau_m = \mathbf{M}(\theta_{\text{act}})\ddot{\theta}_{\text{act}} + \mathbf{C}(\theta_{\text{act}}, \dot{\theta}_{\text{act}}) + \mathbf{G}(\theta_{\text{act}}) - \mathbf{J}(\theta_{\text{act}})^T \mathscr{F}_{\text{tip}} \tag{5}$$

An equation for the motor torques may be derived using equation (5), however, this requires symbolic computation. Additionally, the complexity of the equation grows with the number of degrees of freedom of the robot arm, making it challenging to derive the closed-form analytical expression.

A suitable solution is to use the Recursive Newton-Euler Algorithm (RNEA). This algorithm was implemented exactly as described in [18], and provides a computationally efficient method to evaluate τ_m. To find the torques at the robot's base, the base is imagined as an additional link of the arm that doesn't rotate. The wrench (torque and force compiled into one singular vector) at the base can be found in the same way as the wrench of all other joints using the RNEA. Using the same notation as [18] and letting the subscript 0 denote the robot's base, the wrench $\mathscr{F}_0$ in question is given by:

$$\mathscr{F}_0 = \text{Ad}^T_{T_{1,0}}(\mathscr{F}_1) + \mathscr{G}_0\dot{\mathscr{V}}_0 - \text{ad}^T_{\mathscr{V}_0}(\mathscr{G}_0\mathscr{V}_0) \tag{6}$$

From the wrench, $\mathscr{F}_0 = [\tau_0\ f_0]^T$ the torques τ_0 and forces f_0 at the base can be extracted. Finally, to compute the feedforward torque τ'_f at the center of the satellite body, the components of the wrench are used along with the position of the robot's base with respect to the satellite body r:

$$\tau'_f = \tau_0 - f_0 \times r \tag{7}$$

Sensor-assisted feedforward

Other ways of calculating τ'_f exist, though not without additional hardware or software. For instance, if a 6-axis force and torque sensor were placed at the connection point between the arm and the satellite, the sensor would immediately provide $\mathscr{F}_0$, rendering the RNEA step and equation (6) unnecessary. This method was tested by the authors and provided similar results as RNEA, though the tests were only conducted with a perfect sensor. It is possible that an imperfect sensor could lead to worse performance and reduced stability.

Predictive feedforward

Alternatively, one could simulate the behavior of the vehicle a number of time steps in advance, then use force and torque data gathered from this simulation to inform a second simulation (or a real satellite) exactly as the robot is performing the movement without time delay. In theory, this method is more advantageous than the normal method described previously, which is prone to an unavoidable time delay between the application of the reference trajectory to the robot arm and the measurement of the true trajectory. In simulation, a further time delay exists between the application of any control input torque and the appearance of its effect on satellite attitude.

There is added complexity in structuring a forecast simulation. The results of the forecast can be used to inform control decisions of the real spacecraft; however, because the inputs to both vehicles become different, the forecast state diverges from the ground truth. A specific architecture that constantly restarts the forecast simulation could be employed, however, this was not explored further, and due to this complexity, the idea of a forecast simulation was eventually abandoned by the authors.

GAIN SCHEDULING

As a solution to the second problem identified in *Problems with the basic controller*, the method of Gain Scheduling can be used. This method involves continuous adjustment of the control gains **P** and **D** such that the setpoint tracking performance of the system remains consistent across all manipulator poses. To do this, the overall inertia matrix I of the vehicle, as shown in equation (3) can be recalculated at every timestep, leading to pose-dependent **A** and **B** matrices and hence, pose-dependent gains.

It should be noted that in this research, Gain Scheduling has not been implemented to its full extent as described above. Instead, a simplified approach has been taken in which a small number of inertia matrices I have been pre-computed for specific test poses of the manipulator. These inertia matrices have then been used to pre-compute the constant controllers used for testing.

RESULTS

Based on the improvements described in the earlier sections, an advanced control system was developed. This can be seen in Figure 4. For the theoretical RPO vehicle, the **P** and **D** gains from Equation 4 were tuned based on poles of -1.3 and -0.3, respectively. The behavior of parts of the system was tested and measured in the following 4 scenarios:

1. Tracking: Slew maneuver with a static robot arm.
2. Disturbance: Movement of the robot arm with a static attitude setpoint.
3. Satellite servicing: Combination of the previous 2 scenarios.
4. RSO capture: Space debris capture with a static robot arm and a static attitude setpoint.

Scenario 1

The first scenario was tested by applying an example attitude step input of $(\phi, \theta, \psi) = (1, 1, 0)$ rad, which is first converted to its quaternion representation before being input to the control loop as q_{ref}. Figure 5 shows the evolution of the attitude represented as Euler angles. This visualization is considered impractical by the authors as it may be difficult to interpret when close to representational singularities, furthermore this view makes it difficult to quantify the attitude error.

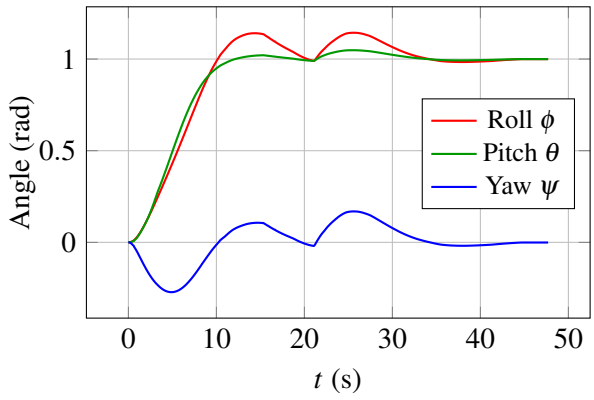

Figure 5. Basic system, Euler angle attitude, slew response of 1 rad for Roll and Pitch

A better representation is the equivalent rotation angle α (shown in Figure 2), which represents the magnitude of the absolute attitude error. Furthermore, it has a physical meaning as it refers to the magnitude of the rotation that would need to be performed around the current equivalent rotation axis in order to perfectly return to the attitude setpoint q_{ref}. This makes it a good metric for assessing performance, as it is a single parameter rather than 3. The same response represented α is shown in Figure 6. The controller used in this case was tuned without regard for the existence of the robot arm, meaning the inertia matrix I used in equation 3 was that of a cube (satellite body) with a uniform density. As observed in the graph, the attitude error is reduced; however, due to the difference between the true inertial properties of the vehicle and the ones used for tuning, the error oscillates before stabilizing.

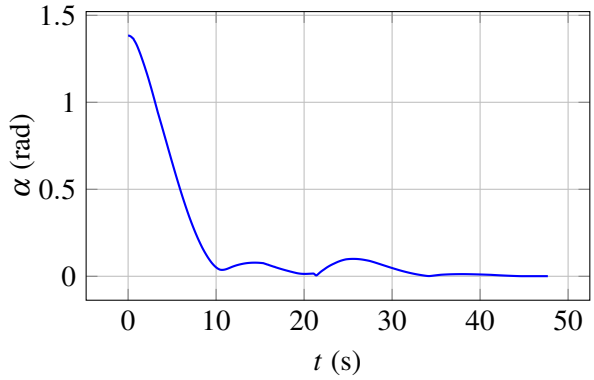

Figure 6. Basic system, Equivalent angle α, slew response of 1 rad for Roll and Pitch.

As an improvement of the first scenario, different controllers that were more closely tuned for specific poses of the robot arm were also tested. In Figure 7, slew maneuvers were performed while keeping the robot arm first in an extended (straight) pose, then in a pose with one joint at 90°, forming an "L". In both cases, the controller used was tuned specifically for the straight pose. As can be seen, the difference that altering the pose makes while maintaining the same controller does not cause a significant difference in the response. Nevertheless, the graphs demonstrate that using the appropriate controller can eliminate the oscillations seen in Figure 6.

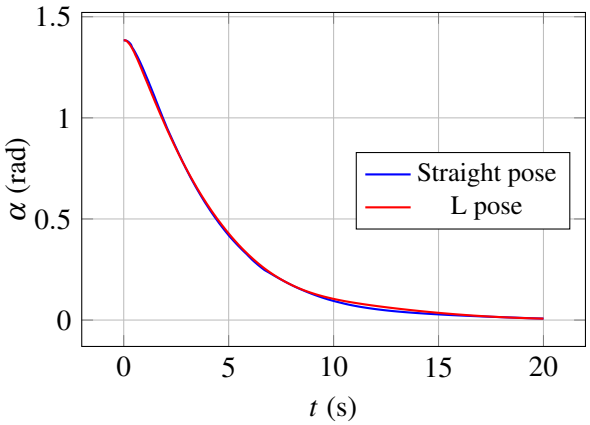

Figure 7. Manipulator poses, straight controller, slew response 1 rad for Roll and Pitch.

The next test, displayed in Figure 8, is almost identical to the previous one, with the difference that the controller used for the L pose robot was now also tuned specifically for this pose. The observed responses now visibly deviate more, highlighting the fact that the behavior change from switching controllers tuned to 2 specific poses is greater than the behavior change from moving the robot itself between said poses.

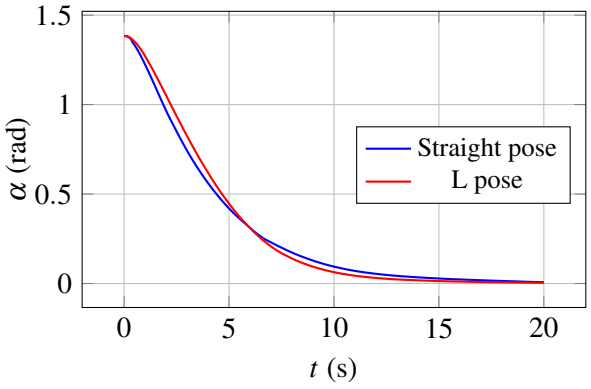

Figure 8. Manipulator poses, adequately tuned controllers, slew response 1 rad for Roll and Pitch.

Scenario 2

Scenario 2 was tested by keeping the attitude setpoint static while providing a trajectory for the robot arm. This trajectory was the same across all tests of the research. The results are shown in Figure 9. Both responses utilized a controller tuned for a uniform cube. As expected, the feedforward element reduces the maximum attitude error during manipulator motion, in this case by approximately 40%. It should be noted that the disturbances are not canceled out completely since the full dynamics of the combined manipulator and satellite system have not been modeled analytically, as has been done in other research [19]. Despite this, the results show that a simplified approach using RNEA may be enough to provide sufficient disturbance rejection depending on the application and specific mission requirements.

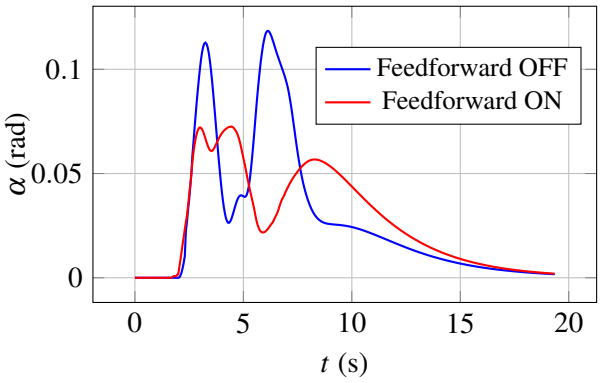

Figure 9. Disturbance performance, basic controller.

The previous test was also conducted for a different set of controllers. In Figure 10 , the ”basic system” refers to a setup tuned for a uniform cube without an active feedforward element. Meanwhile, the ”advanced system” refers to a setup tuned for the specific pose the robot arm is in, in addition to containing an active feedforward element. The difference in attitude error is even more substantial than in Figure 9, and the reduction in maximum attitude error is approximately 63%, emphasizing the advantages of using a gain-scheduled controller.

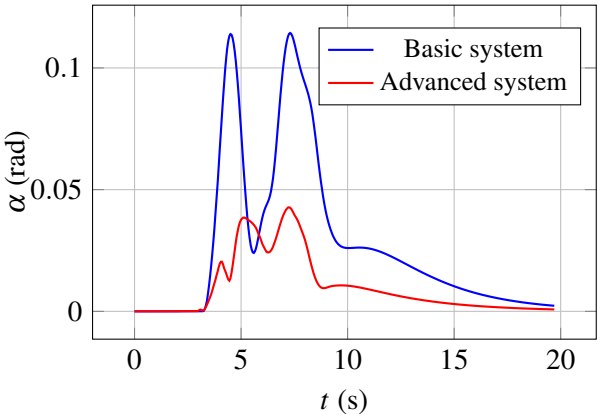

Figure 10. Attitude error during disturbance.

Nevertheless, the improved disturbance rejection comes at the cost of higher actuator efforts. In Figure 11, the total angular momentum stored in the three reaction wheels (RWs) during the same test performed in Figure 10 is graphed. As can be seen, the maximum angular momentum requirement is approximately 163% higher in the advanced system, which is an extremely significant increase. However, it should be noted that the arm trajectory used in the tests was fast and highly dynamic relative to the inertial properties of the satellite. The duration of a manipulator trajectory in a real mission may likely be in the order of minutes rather than seconds, which would greatly decrease dynamic disturbances on the satellite body, thereby decreasing the angular momentum requirements of the actuators.

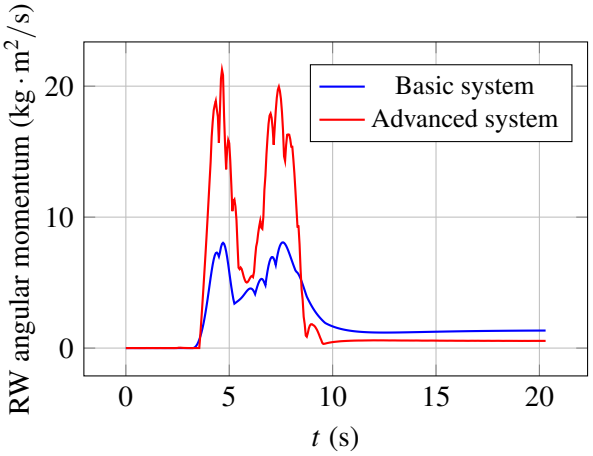

Figure 11. Actuator effort during disturbance.

Scenario 3

Scenarios 3 and 4 are where the performance of the ”advanced system” stands out the most in comparison to the ”basic system”. For Scenario 3, the same attitude step response as described earlier was input into the system to perform a slew maneuver, while the manipulator performs a trajectory movement simultaneously. The results are shown in Figure 12. Due to the presence of a feedforward element in the advanced version of the system, attitude oscillations that were present in the response of the basic system have been almost completely eliminated, resulting in a visibly smoother overall response.

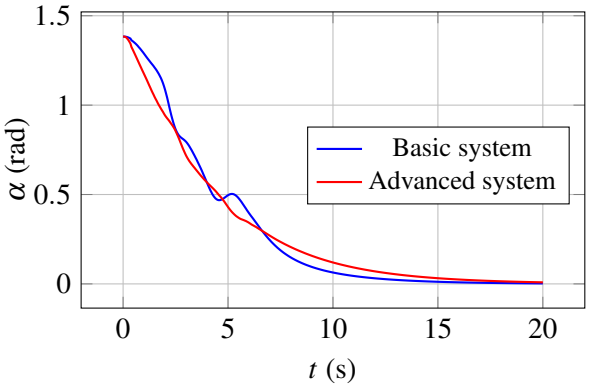

Figure 12. Attitude error during slew and disturbance.

The angular momentum requirements of the Scenario 3 test, displayed in Figure 13, show an increase in maximum angular momentum requirement of approximately 52% from the basic system to the advanced one. This increase is to be expected given the higher complexity of the control objective, but it is nowhere near as high as the one found previously in Scenario 2. Considering the stark difference in actuator requirement increase (163% vs. 52%), the results could indicate that the majority of the control effort is expended in the execution of the slew maneuver rather than the compensation of manipulator disturbances. Nevertheless, this ratio may vary for satellite bodies or manipulators with different inertial properties.

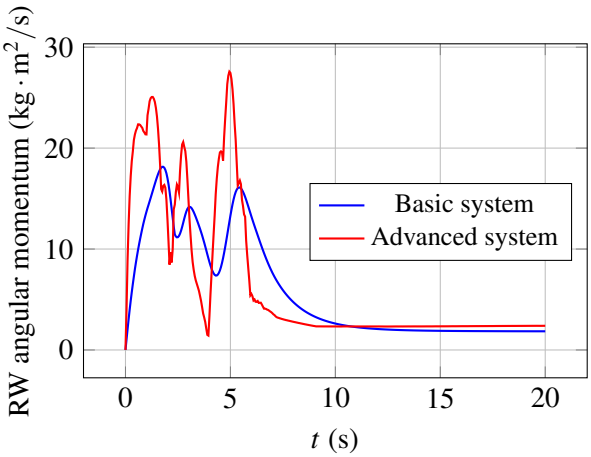

Figure 13. Basic vs. advanced system step response slew, graphing angular momentum.

Scenario 4

In Scenario 4, a 50kg block imitating a piece of space debris was set up to impact the end effector of the robot arm, while moving at a relative linear velocity of 1 m/s at the moment of impact. Upon impact, the debris was programmed to rigidly attach to the end effector to simulate capture. Figure 14 shows the results of this test using two different controllers: one accounting for the debris (the RSO controller) and one disregarding it (the Straight controller). As expected, the capture of the debris causes a rapid increase in attitude error α due to the reaction force exerted on the vehicle. The maximum attitude error during the RSO controller response is 34% lower than that of the Straight controller response.

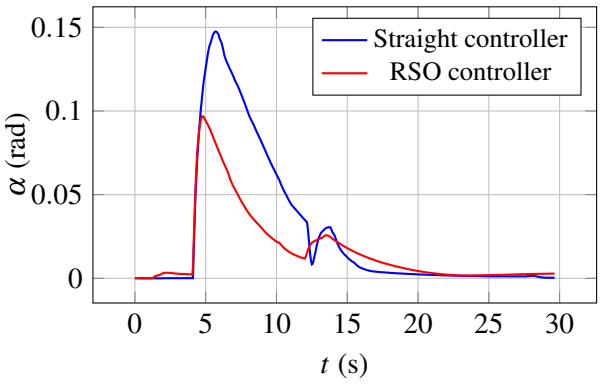

Figure 14. RSO capture, graphing α.

Finally, the angular momentum requirements for the Scenario 4 test are shown in Figure 15. Somewhat surprisingly, the requirements for both the RSO controller case and the Straight controller case are almost identical, despite the RSO controller having more aggressive gains due to it being tuned accounting for a debris located far from the center of mass of the satellite body. Furthermore, when the system settles into steady-state, a constant angular momentum of approximately 45 $\mathrm{kg} \cdot \mathrm{m}^2/\mathrm{s}$ remains stored in the reaction wheels in both cases. This is due to the transfer of linear momentum from the moving debris to the vehicle, part of which is absorbed by the reaction wheels while the rest remains as a constant linear momentum of the satellite-manipulator-debris system after capture.

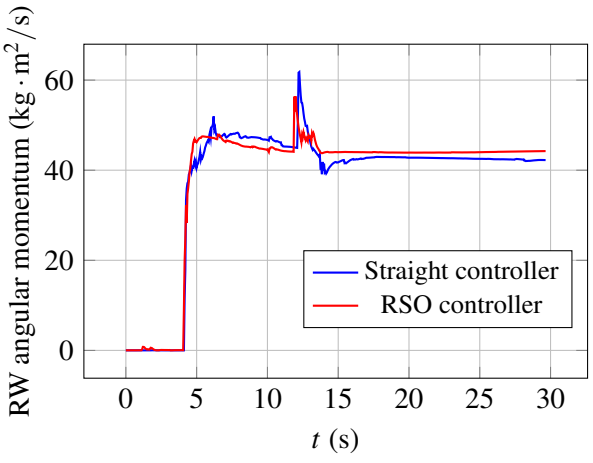

Figure 15. RSO capture, graphing angular momentum.

CONCLUSION

Based on experiments performed with the systems, several insights into the designs of RPO vehicle control systems have been revealed. The first insight is related to adjusting controller gains based on the pose of the robotic manipulator. Adjusting gains ensures a constant response across the whole robotic workspace. The model that is used to select the initial gains changes as the manipulator changes pose, however, it is important to quantify whether these changes prevent a static controller from achieving mission requirements. The potential pitfall in adjusting controller gains is that an unexpectedly high or low gain could be selected, which does not meet mission requirements.

A feedforward controller causes the actuator effort required to increase significantly. The disturbances caused by the motion of the arm are related to the speed of the trajectory and the inertia properties of the manipulator. When mission planning for RPO, it is important to consider different parts of the mission with their own specific requirements, for example, is it possible to have a lower pointing requirement while the robotic manipulator is executing a trajectory? In cases where both a low pointing error and manipulator motion is necessary, feedforward based on sensor measurements is a good strategy. Furthermore, the simplified feedforward algorithm implemented in our paper, which does not require time-consuming analytical modeling of the manipulator, works well to counter disturbances from trajectories with high velocity (in the order of mm/second). This performance may be even better for slow trajectories (in the order of mm/minute or less).

RSO capture poses several problems for ADCS design. Firstly, adjusting controller gains is very important, as the addition of mass to the system changes the inertial parameters, which also alter the performance of the system. In the case of cooperative objects, from the results of our study, it is best to utilize a controller tuned for the combined inertial parameters when performing the initial attitude correction. This poses a problem for non-cooperative objects where the parameters are unknown before col-

lision. To solve this problem, additional sensors will be required to perform an estimation of these parameters. Furthermore, an even more important factor for deciding whether to perform a capture operation is the overall momentum capacity of the actuators of the ADCS. By capturing an RSO, there will be a momentum transfer. Based on the results of the experiments within this study, the geometry of how the capture is performed directly influences the actuator effort required to stabilize the vehicle post-collision. It is important to position the robotic manipulator to minimize the angular momentum transfer from the collision. In our study, this meant that the capture should be executed in such a way as to minimize the orthogonal distance between the capture axis of the incoming RSO and the center of the spacecraft mass. Figure 16 depicts the theoretical vehicle preparing to capture an RSO.

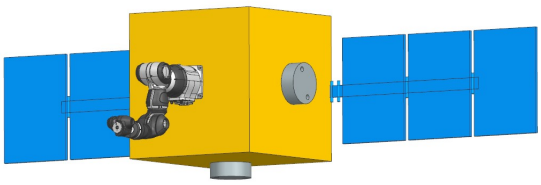

Figure 16. Theoretical RPO vehicle undergoing RSO capture

FUTURE WORK

Our study suggests that future studies into the control policy for an RPO vehicle must consider how the attitude control system interacts with additional sensors for the estimation of inertial properties of capture targets. Furthermore, an advanced motion planner for the robotic arm could be developed to include the capture plans into the attitude control loop, and to make decisions on whether a capture should take place based on RSO parameter estimates, trajectory planning, and the current state of the actuators. Just as important as the capture strategy is the actuator desaturation strategy. If an RSO capture inputs enough angular momentum into the system, the attitude control system may not be suited to ensure controllability of the spacecraft. Determining an adequate desaturation strategy is an important step in the RSO capture workflow.

ACKNOWLEDGMENTS

The authors would like to extend a warm thank you the following academic supervisors at the KU Leuven University for supporting this research: Tim Catoor, Patrick Geenen. Additionally the authors would like to express their gratitude to the postdoctoral researcher of KU Leuven University: Mathias Bos for his support and invaluable feedback on this research.

BIBLIOGRAPHY

1. *Attitude Determination and Control 1.0rc2 documentation*. [Online; accessed 13. Apr. 2025]. Feb. 2021. URL: https://nanostar-project.gitlab.io/main/source/preliminary-design/adcs.html.
2. Douglas Cline. *13.17: Euler's equations of motion for rigid-body rotation*. Mar. 2021. URL: https://phys.libretexts.org/Bookshelves/Classical_Mechanics/Variational_Principles_in_Classical_Mechanics_(Cline)/13%3A_Rigid-body_Rotation/13.17%3A_Eulers_equations_of_motion_for_rigid-body_rotation.
3. M. Lazzarin et al. "EUCLID satellite: Sloshing model development through computational fluid dynamics". In: *Aerospace Science and Technology* 36 (2014), pp. 44–54. ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2014.03.015. URL: https://www.sciencedirect.com/science/article/pii/S1270963814000637.
4. Jia Chen et al. "Post-capture Angular Momentum Management of Space Robot with Controllable Damping Joints". In: *2019 IEEE 2nd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*. 2019, pp. 638–642. DOI: 10.1109/AUTEEE48671.2019.9033432.
5. Angel Flores-Abad et al. "A review of space robotics technologies for on-orbit servicing". In: *Progress in Aerospace Sciences* 68 (2014), pp. 1–26. ISSN: 0376-0421. DOI: https://doi.org/10.1016/j.paerosci.2014.03.002.

URL: https://www.sciencedirect.com/science/article/pii/S0376042114000347.

6. Zhijie Liu, Jinkun Liu, and Lijun Wang. "Disturbance observer based attitude control for flexible spacecraft with input magnitude and rate constraints". In: *Aerospace Science and Technology* 72 (2018), pp. 486–492. ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2017.11.036. URL: https://www.sciencedirect.com/science/article/pii/S1270963817304078.

7. Zengbo Liu, Yukai Zhu, and Jianzhong Qiao. "Composite anti-disturbance position and attitude control for spacecrafts with parametric uncertainty and flexible vibration". In: *Chinese Journal of Aeronautics* 35.12 (2022), pp. 242–252. ISSN: 1000-9361. DOI: https://doi.org/10.1016/j.cja.2022.01.005. URL: https://www.sciencedirect.com/science/article/pii/S100093612200005X.

8. R. Rodrigues et al. "Modeling, robust control synthesis and worst-case analysis for an on-orbit servicing mission with large flexible spacecraft". In: *Aerospace Science and Technology* 129 (2022), p. 107865. ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2022.107865. URL: https://www.sciencedirect.com/science/article/pii/S1270963822005399.

9. Aureliano Rivolta, Paolo Lunghi, and Michèle Lavagna. "GNC robotics for on orbit servicing with simulated vision in the loop". In: *Acta Astronautica* 162 (2019), pp. 327–335. ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2019.06.005. URL: https://www.sciencedirect.com/science/article/pii/S0094576518308968.

10. José L. Ramón, Jorge Pomares, and Leonard Felicetti. "Direct visual servoing and interaction control for a two-arms on-orbit servicing spacecraft". In: *Acta Astronautica* 192 (2022), pp. 368–378. ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2021.12.045. URL: https://www.sciencedirect.com/science/article/pii/S0094576521006895.

11. Khush Thakor et al. "Results From On-Orbit Operation of CubeSat-Scale Robotic Arms on the International Space Station". In: *DigitalCommons@USU* (2024). URL: https://digitalcommons.usu.edu/smallsat/2024/all2024/65.

12. *Walking Manipulator*. [Online; accessed 14. Apr. 2025]. June 2023. URL: https://www.spaceapplications.com/products/walking-manipulator.

13. *Inchworm Robot | GITAI*. [Online; accessed 15. Apr. 2025]. Oct. 2023. URL: https://gitai.tech/inchworm-robot.

14. Bartlomiej Koszek and Alvaro Roman Sanchez. "Development of digital twin of CubeSat equipped with robotic manipulator". In: (Jan. 2025). https://orcid.org/0009-0006-6965-774X. URL: https://spaceworkshop.fi/program2025.html.

15. *Arcsec: Sagitta Datasheet V2.1*. [Online; accessed 16. Apr. 2025]. Mar. 2025. URL: https://www.arcsec.space/sagitta.

16. Natalia Piotrowski. "Implementation and Characterization of Commercial Off-The-Shelf Inertial Measurement Units for the Attitude Determination System of the MOVE-III CubeSat". In: *DigitalCommons@USU* (2022). URL: https://digitalcommons.usu.edu/smallsat/2022/all2022/1.

17. Emil Fresk and George Nikolakopoulos. "Full quaternion based attitude control for a quadrotor". In: *2013 European Control Conference (ECC)*. 2013, pp. 3864–3869. DOI: 10.23919/ECC.2013.6669617.

18. Kevin M. Lynch and Frank C. Park. "Newton-Euler Inverse Dynamics Algorithm". In: *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017, pp. 292–293. ISBN: 9781107156302. URL: https://hades.mech.northwestern.edu/images/2/25/MR-v2.pdf.

19. Markus Wilde et al. “Equations of Motion of Free-Floating Spacecraft-Manipulator Systems: An Engineer’s Tutorial”. In: *Frontiers in Robotics and AI* Volume 5 - 2018 (2018). ISSN: 2296-9144. DOI: 10 . 3389 / frobt . 2018 . 00041. URL: https : //www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2018.00041.