

# **Field and Service Robotics: Final Project**

Comparative Analysis of Centralized and Decentralized Control for an UAM

Acampora Giulio: P38000258  
Del Sorbo Alessandra: P38000289

## Contents

<b>1 System Modeling and Parameters</b>	<b>3</b>
1.1 Manipulator Kinematics: Denavit-Hartenberg Parameters . . . . .	4
1.2 Coordinate Frame Convention . . . . .	4
<b>2 Centralized Control of an UAM</b>	<b>5</b>
2.1 Dynamic Model of an UAM . . . . .	5
2.2 Cartesian Impedance Control . . . . .	5
2.2.1 Outer Loop: UAV Translational Control . . . . .	6
2.2.2 Geometric Reference Generation for the Inner Loop . . . . .	6
2.2.3 Inner Loop: UAV Attitude and Manipulator Control . . . . .	7
2.3 Simulations - Disturbance Rejection . . . . .	7
2.3.1 Analysis of Case A: Rigid vs. Compliant Behavior . . . . .	7
2.3.2 Analysis of Case B: External Wrench on Manipulator . . . . .	9
2.4 Conclusion . . . . .	11
<b>3 Decentralized Control of an UAM</b>	<b>12</b>
3.1 UAV Passivity-Based Controller . . . . .	12
3.1.1 Momentum-Based Disturbance Estimator . . . . .	12
3.1.2 Outer Loop: Position Control . . . . .	12
3.1.3 Inner Loop: Attitude Control . . . . .	13
3.2 Manipulator Cartesian Impedance Controller . . . . .	13
3.3 Wrench Transformation . . . . .	13
3.4 Simulations - Disturbance Rejection . . . . .	14
3.4.1 Controller Gains . . . . .	14
3.4.2 Analysis of Case A: Rigid vs. Compliant Behavior . . . . .	14
3.4.3 Analysis of Case B: External Wrench on Manipulator . . . . .	15
3.5 Conclusion . . . . .	16
<b>4 Comparative Analysis of Control Strategies</b>	<b>17</b>
4.1 Performance . . . . .	17
4.2 Conclusion . . . . .	18
<b>5 Comparative Analysis with moving Manipulator trajectory</b>	<b>18</b>
5.1 End-Effector Performance . . . . .	19
5.2 UAV Performance . . . . .	19
5.3 Conclusion . . . . .	20
<b>6 Comparative Analysis with moving Manipulator with a Payload</b>	<b>21</b>
6.1 End-Effector and Quadrotor Position Error . . . . .	22
6.2 Conclusion . . . . .	23

This document contains the Final Project of the Field and Service Robotics class.

## Introduction - Comparative Analysis of Centralized and Decentralized Control for an UAM

Unmanned Aerial Manipulators (UAMs), which integrate multirotor UAVs with robotic arms, present significant control challenges due to the strong dynamic coupling between the aerial platform and the manipulator. This report presents the implementation and comparative analysis of two distinct control architectures for a UAM: centralized and decentralized.

- The **centralized strategy** employs a single controller that manages the entire UAM based on its complete, coupled dynamic model.
- The **decentralized strategy** treats the UAV and manipulator as separate subsystems, where the UAV's controller actively compensates for the manipulator's motion as an external disturbance. This disturbance is quantified in real time using a momentum-based wrench estimator.

The objective of this report is to provide a detailed technical breakdown of each implementation, evaluate their performance based on simulation results, and present a conclusive analysis of their respective advantages and limitations.

Several simulation scenarios have been considered to assess the effectiveness of the proposed control strategies:

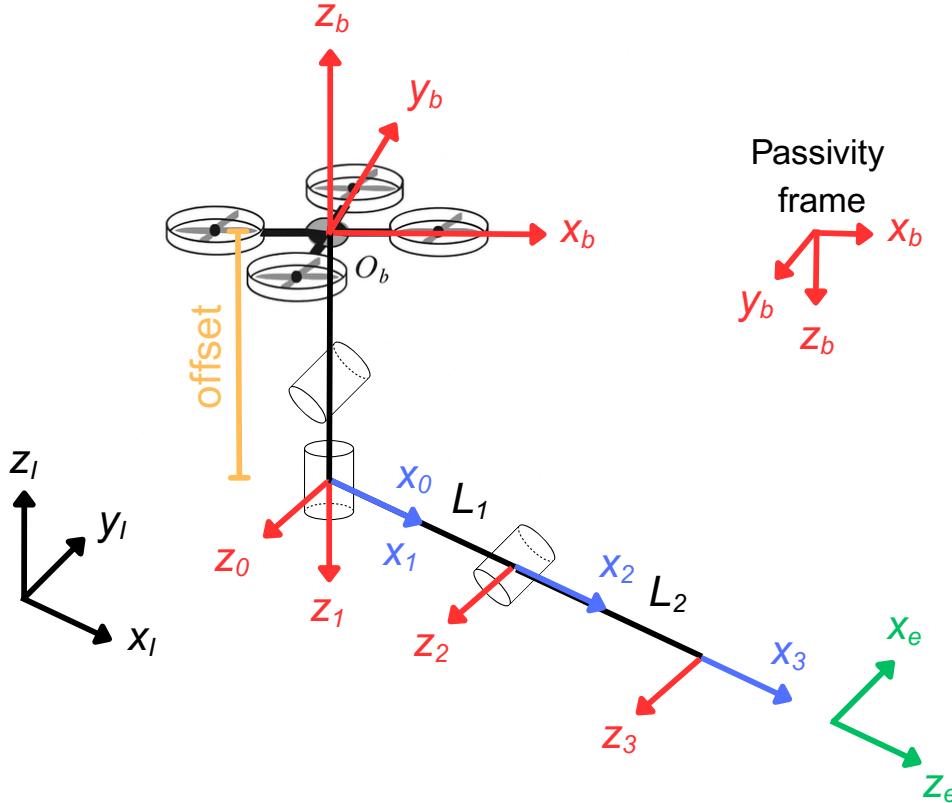
- **Disturbance rejection:** evaluation of the UAM behavior under external perturbations, such as wind gusts or collisions with obstacles, while maintaining quasi-static hovering stability.
- **Trajectory execution:** analysis of the manipulator's ability to follow a prescribed trajectory while the UAV sustains a stable hovering position.
- **Load handling:** assessment of the system performance when the manipulator interacts with or carries an external load.

## 1 System Modeling and Parameters

The UAM model consists of a quadrotor UAV and a 3-DOF manipulator, as shown in Fig. 1. The robotic manipulator is composed of two links and three revolute joints, with the first two axes intersecting at a common point. The physical parameters, as defined in `UAM_initialization.m`, are listed below:

Parameter	Symbol	Value	Description
UAV Mass	$m_b$	2.0 kg	Mass of the quadrotor body
UAV Inertia Matrix	$H_b$	$\text{diag}([1.24, 1.24, 2.48]) \text{ m}^2\text{kg}$	Inertia tensor of the quadrotor
Manipulator Link 1 Mass	$m_{l1}$	0.049 kg	Mass of the first moving link
Manipulator Link 2 Mass	$m_{l2}$	0.05 kg	Mass of the second moving link
Manipulator Link 1 Length	$l_1$	0.15 m	Length of the first moving link
Manipulator Link 2 Length	$l_2$	0.05 m	Length of the second moving link
Manipulator Offset	offset	0.1 m	Vertical distance from UAV center to arm base

**Table 1:** Table of parameters for the quadrotor and manipulator.



**Figure 1:** UAV structure with all the frames.

## 1.1 Manipulator Kinematics: Denavit-Hartenberg Parameters

The forward kinematics of the 3-DOF manipulator is modeled using the standard Denavit-Hartenberg (D-H) convention. The specific D-H parameters used in this model are:

Link	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	0	$\pi/2$	0	$q_1$
2	$l_1$	$-\pi/2$	0	$q_2$
3	$l_2$	0	0	$q_3$

**Table 2:** D-H table for the Manipulator.

These parameters are used within the `Direct_kinematics` function to compute the end-effector's position relative to the manipulator's base.

## 1.2 Coordinate Frame Convention

A critical implementation detail is the use of two different body-fixed coordinate frames for the UAV (you can see this in Fig. 1):

- **Impedance Frame (Z-Up):** The centralized impedance controller uses a standard robotics frame where the z-axis points upward.

- **Passivity Frame (Z-Down):** The passivity-based UAV controller implemented needs a frame where the z-axis points downward, aligned with gravity. This convention is also adopted in the decentralized UAV planner.

A rotational transformation matrix,  $R_{\text{pass\_imp}}$ , representing a 180-degree rotation about the x-axis, is defined to convert vectors between these frames, ensuring consistency across different parts of the control architecture.

## 2 Centralized Control of an UAM

The centralized control strategy treats the UAM as a single, fully coupled dynamic system.

### 2.1 Dynamic Model of an UAM

The dynamic model has been derived using the Euler–Lagrange formulation, as described in [1]. Consider

$$\xi = [p_b^T \quad \eta_b^T \quad q^T]^T$$

as the vector of generalized coordinates, where  $p_b = [x \ y \ z]^T$  represents the absolute position of the UAV body frame with respect to the inertial frame,  $\eta_b = [\phi \ \theta \ \psi]^T$  are the roll–pitch–yaw Euler angles describing the UAV attitude, and  $q = [q_1 \ q_2 \ q_3]^T$  are the manipulator’s joint variables. The dynamics of the whole system can then be written in compact matrix form as

$$B(\xi) \ddot{\xi} + C(\xi, \dot{\xi}) \dot{\xi} + g(\xi) = u + u_{\text{ext}}, \quad (1)$$

where  $u$  represents the generalized input forces and  $u_{\text{ext}}$  accounts for the effects of external generalized forces at the joint level.

The inertia matrix  $B(\xi)$  is  $9 \times 9$ , symmetric, and positive definite. Its block elements are defined in the referenced paper [1] and in the provided code. The Coriolis matrix  $C(\xi, \dot{\xi}) \in \mathbb{R}^{9 \times 9}$  has generic element

$$c_{ij} = \sum_{k=1}^9 \frac{1}{2} \left( \frac{\partial b_{ij}}{\partial \xi_k} + \frac{\partial b_{ik}}{\partial \xi_j} - \frac{\partial b_{jk}}{\partial \xi_i} \right) \dot{\xi}_k, \quad (2)$$

and has been computed using an external MATLAB script `calculate_C_matrix`.

Finally, the gravity vector is defined as

$$g(\xi) = \left( \frac{\partial U(\xi)}{\partial \xi} \right)^T, \quad (3)$$

where  $U(\xi)$  is the potential energy of the system.

### 2.2 Cartesian Impedance Control

Now that the fully coupled dynamic system has been derived, the next step is to implement the controller. In particular, a revised version of the Cartesian Impedance Control described in [1] has been considered. The chosen Cartesian variables for the control are the pose of the aerial vehicle and the position of the manipulator end-effector:

$$x = [p_b^T \quad \eta_b^T \quad p_e^T]^T,$$

where  $p_b$  is the UAV position,  $\eta_b$  the Euler angles, and  $p_e$  the end-effector position.

By denoting the tracking error as

$$\tilde{x} = x_d - x,$$

the control law can be defined as

$$u = g(\xi) + J(\xi)^T (B_x \ddot{x}_d + C_x \dot{x}_d + K_D \dot{\tilde{x}} + K_P \tilde{x}), \quad (4)$$

where  $K_P, K_D \in \mathbb{R}^{9 \times 9}$  are symmetric and positive definite matrices representing the desired stiffness and damping, respectively. The matrices  $B_x$  and  $C_x$ , which represent the inertia and Coriolis terms with respect to the Cartesian variables  $x$ , are defined as

$$B_x = J(\xi)^{-T} B(\xi) J(\xi)^{-1}, \quad (5)$$

$$C_x = J(\xi)^{-T} (C(\xi, \dot{\xi}) - B(\xi) J(\xi)^{-1} \dot{J}(\xi)) J(\xi)^{-1}, \quad (6)$$

where the Jacobian  $J(\xi) \in \mathbb{R}^{9 \times 9}$  is defined such that

$$\dot{x} = J(\xi) \dot{\xi}.$$

A critical challenge arises from the physical nature of the quadrotor base: it is an underactuated system. To overcome this, the controller, while centralized in its knowledge of the full system dynamics, is implemented with a logical inner-outer loop structure. This design explicitly acknowledges that the UAV's translational position ( $x, y, z$ ) and yaw ( $\psi$ ) are the primary commandable outputs, while the roll ( $\phi$ ) and pitch ( $\theta$ ) angles are not independent trajectories but are instead consequences of the desired translational motion.

### 2.2.1 Outer Loop: UAV Translational Control

The outer loop is exclusively responsible for the UAV's 3-DOF translational motion. It computes a virtual acceleration command  $\mu$  that represents the total desired force vector acting on the UAM's center of mass that must be exerted by the UAM to track its position trajectory. The control law is a form of impedance control that uses the full system dynamics:

$$\mu_d = g_p + J_p^T (B_x \ddot{p}_{b,d} + C_x \dot{p}_{b,d} + K_d \dot{\tilde{p}}_b + K_p \tilde{p}_b) \quad (7)$$

This control law utilizes the specific partition of the full 9x9 system Jacobian,  $J_p$ , that maps the UAV's translational velocities ( $\dot{p}_b$ ) to the task-space velocities of the complete system. Similarly, it uses the corresponding partitions of the system's task-space inertia ( $B_x$ ) and Coriolis ( $C_x$ ) matrices. This formulation allows the outer loop to calculate the force vector  $\mu_d$  required to produce the desired translational motion, accounting for the dynamic effects of the entire system on the UAV's position.

### 2.2.2 Geometric Reference Generation for the Inner Loop

This stage translates the virtual force command  $\mu_d$  from the outer loop into actionable physical references for the inner loop. The magnitude of the force vector,  $u_T = \|\mu_d\|$ , directly determines the **total thrust** that must be generated by the UAV's propellers.

The direction of the force vector defines the primary axis of the desired attitude,  $z_{b,d} = \mu_d / \|\mu_d\|$ . While the desired yaw angle  $\psi_d$  is provided by the planner, the desired roll and pitch angles are calculated as

$$\phi_d = \arcsin\left(\frac{-\bar{\mu}_2 \cos(\psi_d) + \bar{\mu}_1 \sin(\psi_d)}{u_T}\right), \quad (8)$$

$$\theta_d = \arctan\left(\frac{\bar{\mu}_1 \cos(\psi_d) + \bar{\mu}_2 \sin(\psi_d)}{\bar{\mu}_3}\right), \quad (9)$$

in order to align the UAV's body  $z$ -axis with the desired force direction. This ensures that the thrust is oriented correctly to achieve the commanded acceleration.

Finally, these references are discretized and filtered to obtain the derivatives  $\dot{\eta}_d$  and  $\ddot{\eta}_d$ .

### 2.2.3 Inner Loop: UAV Attitude and Manipulator Control

The inner loop is a unified controller for the remaining 6 DOFs (3 for UAV attitude, 3 for the manipulator). The state vector for this subsystem is defined as  $x_2 = [\eta_b^T, p_e^T]^T$ . This vector concatenates the UAV's orientation angles and the end-effector's Cartesian position, representing the variables that the inner loop must simultaneously track. The inner loop's function is to compute the actuator commands (UAV attitude torques and manipulator joint torques) necessary to make the actual state track the desired state, which includes the desired attitude  $\eta_d$  and the desired end-effector trajectory  $p_{e,d}$  (from the planner). The controller computes a single 6x1 control vector,  $u_{orient,manip}$ , using a unified impedance law. The logic is to use the partitions of the system matrices that govern the coupled rotational and manipulation dynamics:

$$u_{orient,manip} = g_{orient,manip} + J_{orient,manip}^T(B_x \ddot{x}_{2,d} + C_x \dot{x}_{2,d} + K_d \dot{\tilde{x}}_2 + K_p \tilde{x}_2) \quad (10)$$

Here, by using these specific sub-matrices, the controller explicitly manages the dynamic coupling between the UAV's rotation and the manipulator's movement. This allows it to calculate the precise torques needed for coordinated tracking of both attitude and manipulation objectives.

## 2.3 Simulations - Disturbance Rejection

To evaluate the behavior of the controlled UAM under external perturbations, two simulated case studies, inspired by the referenced paper [1], were carried out. In both cases, the total simulation time was set to 20 s, with the objective of maintaining a hovering condition. The initial values of the system generalized coordinates were defined as

$$\xi = [0 \ 0 \ 2 \ 0 \ 0 \ 0 \ -\pi/2 \ 0 \ \pi/2]^T.$$

In **Case A**, an external force simulating a wind disturbance was modeled as a sine wave with frequency  $\pi/2$  rad/s and amplitude 1 N. In order to model the wind effect blowing in only one direction, the force assumes zero values during the negative half-period of the wave. This force was applied along the UAV body-frame  $x_b$  axis ( $\Sigma_b$ ) during the first 10 seconds of the simulation. Two different control configurations were tested:

- **Case A<sub>1</sub>**: representing a highly stiff (rigid) behavior.
- **Case A<sub>2</sub>**: representing a compliant behavior.

**Case B** was designed to analyze the system's response to an external wrench applied directly to the manipulator. In addition to the wind disturbance considered in Case A, a constant force of 0.5 N was applied along the manipulator end-effector's approach axis for the entire 20 s simulation, simulating contact with a wall.

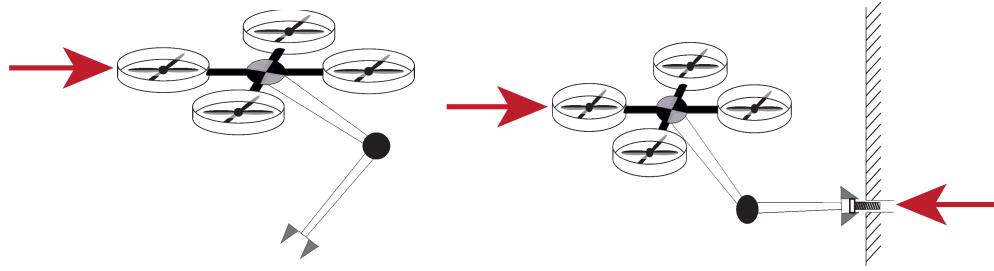
Both case studies are schematically illustrated in Fig. 2. The performance of each case was evaluated by analyzing the time evolution of three error norms: the quadrotor position error norm ( $\|e_{p_b}\|$ ), the quadrotor orientation error norm ( $\|e_{\eta_b}\|$ ), and the end-effector position error norm ( $\|e_{p_e}\|$ )

### 2.3.1 Analysis of Case A: Rigid vs. Compliant Behavior

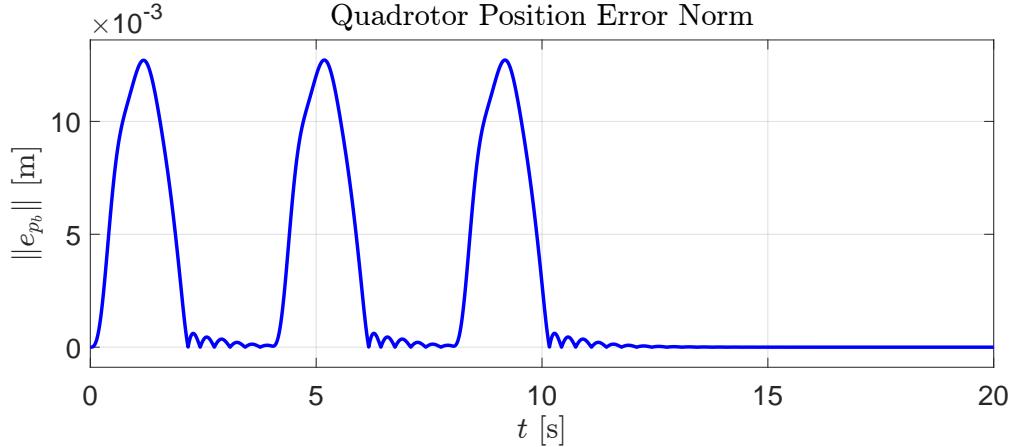
The results for Case A highlight the fundamental difference between the rigid ( $A_1$ ) and compliant ( $A_2$ ) control strategies. In both cases, the external wind force was applied to the UAV. For Case  $A_1$ , the rigid behavior was achieved with a stiffness gain of  $K_p = 80I_9$  and damping ratio  $\zeta$  tuned to 0.8, computing the elements

$$K_D(i,i) = 2\zeta\sqrt{K_P(i,i)}.$$

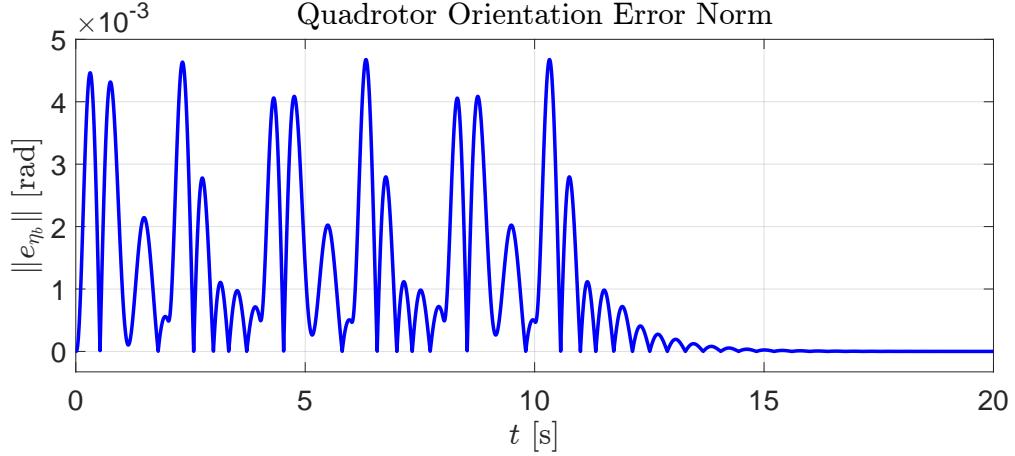
This rigid behavior is evident from the quick convergence and low magnitude of the error norms after the external force is removed. As seen in Figure 3, the quadrotor position error norm peaks at approximately



**Figure 2:** On the left, a schematic representation of the simulated case study A. On the right, a schematic representation of the simulated case study B.

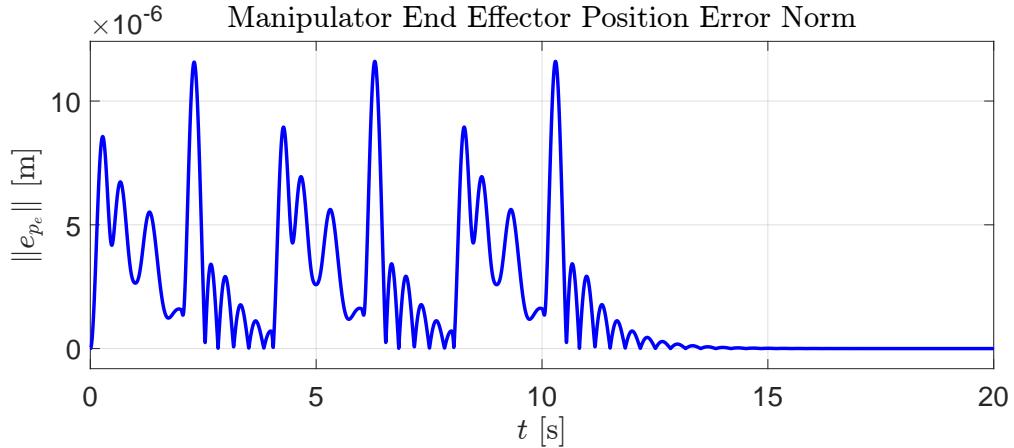


**Figure 3:** Quadrotor Position Error Norm for centralized control for Case A<sub>1</sub> (Rigid Behavior). This plot shows the time evolution of the quadrotor's position error norm when the system is controlled with a rigid behavior ( $K_p = 80I_9$  and damping ratio tuned to 0.8) under external forces applied to the UAV.



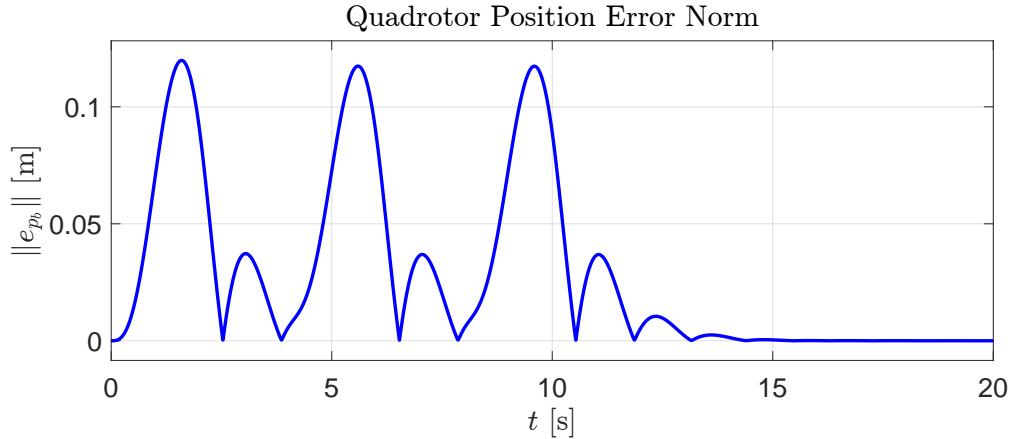
**Figure 4:** Quadrotor Orientation Error Norm for centralized control for Case A<sub>1</sub> (Rigid Behavior). This plot shows the time evolution of the quadrotor's orientation error norm under the rigid control strategy ( $K_p = 80I_9$  and damping ratio tuned to 0.8).

0.012 m before rapidly settling to zero after the force ceases. Similarly, as shown in Figure 4, the quadrotor orientation error norm reaches a maximum of about  $4 \times 10^{-3}$  rad and stabilizes swiftly. The end-effector position error norm (Figure 5) is particularly small, with a peak value on the order of  $10^{-5}$  m, which is



**Figure 5:** End-Effector Position Error Norm for centralized control for Case A<sub>1</sub> (Rigid Behavior) - This plot displays the end-effector position error norm for the rigid control case ( $K_p = 80I_9$  and damping ratio tuned to 0.8).

a direct consequence of the controller's stiff nature. In contrast, Case A<sub>2</sub>, which simulates a compliant

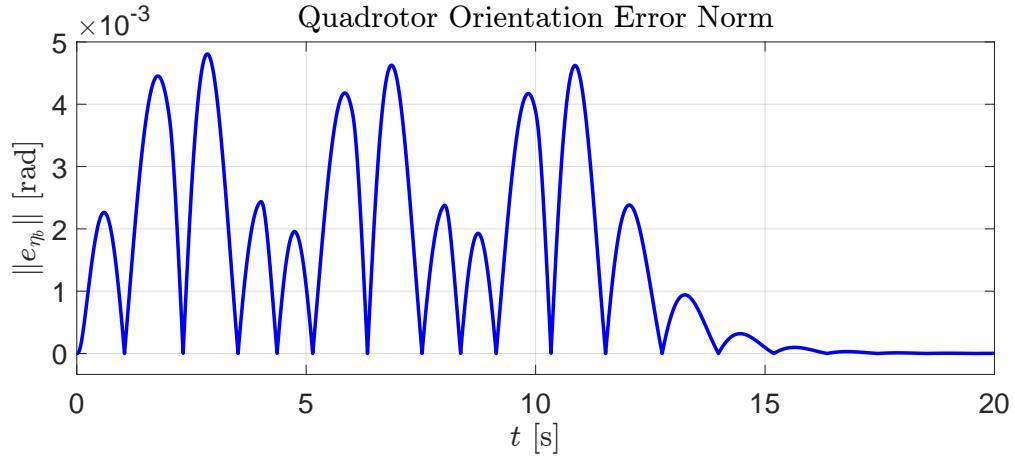


**Figure 6:** Quadrotor Position Error Norm for centralized control for Case A<sub>2</sub> (Compliant behavior). This plot shows the time evolution of the quadrotor's position error norm when the system is controlled with a compliant control behavior ( $K_p = 10I_9$  and damping ratio tuned to 0.5) under external forces applied to the UAV.

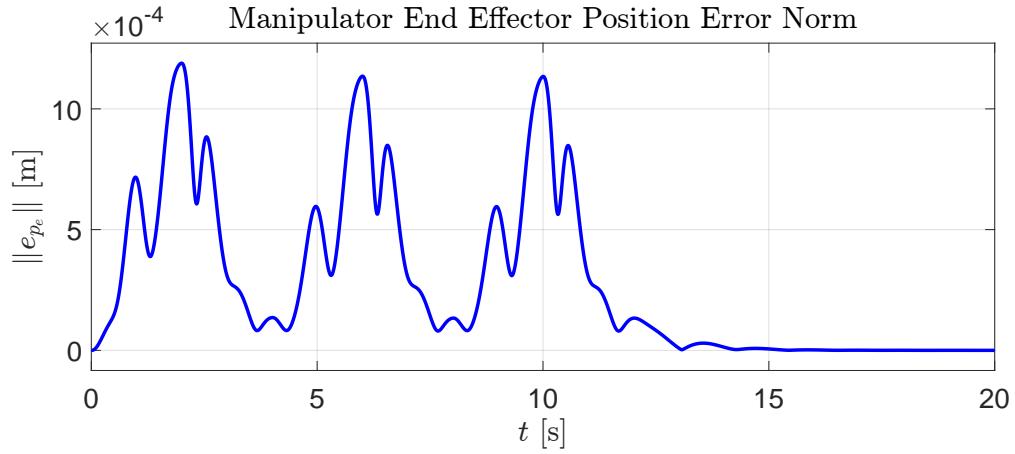
control law, was implemented with a lower stiffness gain of  $K_p = 10I_9$  and damping ratio tuned to 0.5 for each component. This case exhibits significantly larger transient errors due to the controller's design to be compliant to the external force. The peak quadrotor position error norm (Figure 6) is approximately 0.12 m, an order of magnitude higher than in Case A<sub>1</sub>. This is a direct and expected result of the control being designed to comply with the external force, leading to a larger initial displacement. Similarly, the quadrotor orientation and end-effector position error norms (Figures 7 and 8) show larger peaks, reaching approximately  $4.7 \times 10^{-3}$  rad and  $1.2 \times 10^{-3}$  m, respectively. Despite these larger initial deviations, all error norms successfully converge to zero after the external force is removed at 10s, demonstrating the stability of the compliant control.

### 2.3.2 Analysis of Case B: External Wrench on Manipulator

Case B presents a distinct set of results, characterized by two external disturbances. First, the same windy situation from Case A was simulated, with the sine wave force acting for the first 10 seconds. In addition,

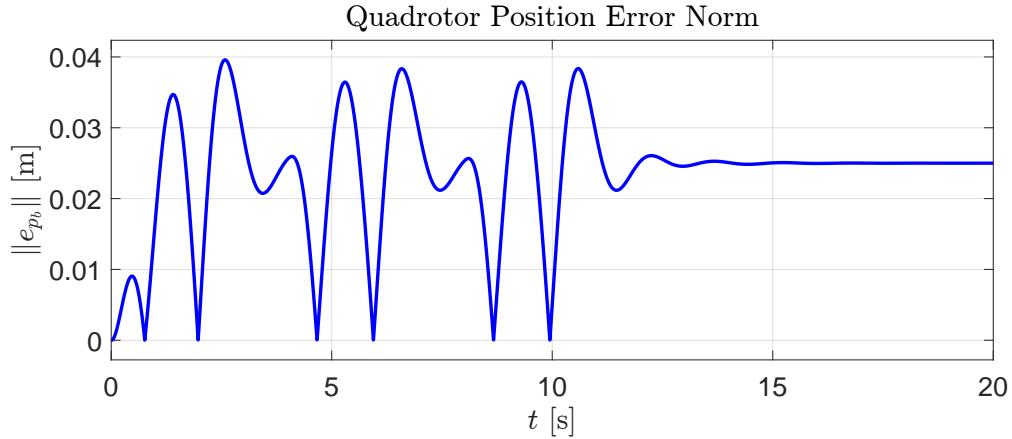


**Figure 7:** Quadrotor Orientation Error Norm for centralized control for Case A<sub>2</sub> (Compliant behavior). This plot shows the time evolution of the quadrotor's orientation error norm under the compliant control strategy ( $K_p = 10I_9$  and damping ratio tuned to 0.5).

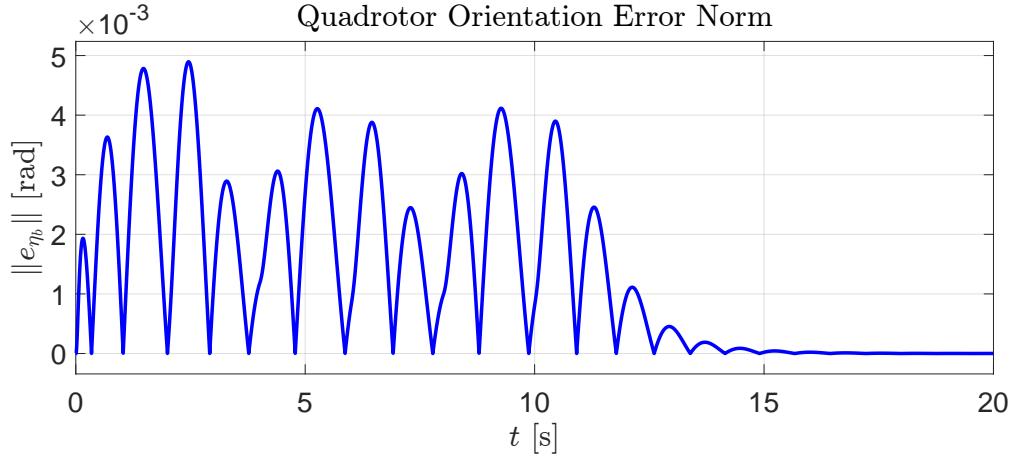


**Figure 8:** End-Effector Position Error Norm for centralized control for Case A<sub>2</sub> (Compliant Behavior) - This plot displays the end-effector position error norm for the compliant control case ( $K_p = 10I_9$  and damping ratio tuned to 0.5).

a constant force applied along the end-effector axis for the entire 20s simulation, simulating contact with a wall. For this scenario, the manipulator end-effector's pose was chosen to be rigid, and the aerial vehicle's pose was chosen to be compliant. The controller was configured with  $K_p = \text{diag}(10I_6, 100I_3)$ , a damping ratio of 0.5 for the UAV components, and 0.85 for the arm end-effector components. As shown in Figure 9, the quadrotor position error norm for Case B shows a highly oscillatory behavior for the first 10 seconds, with a peak amplitude of approximately 0.04 m, directly correlating with the applied sine wave force. After the wind force ceases, the oscillations dampen, and the error settles to a non-zero steady-state value of around 0.025 m. This non-zero error is due to the constant contact force, which prevents the quadrotor from reaching its initial hovering position. The quadrotor orientation error norm (Figure 10) also shows a distinct oscillatory pattern for the first 10 seconds with peaks reaching approximately  $5 \times 10^{-3}$  rad, before dampening and converging to zero. Instead of analyzing the error norm for the end effector, Figure 11 plots the individual components ( $e_x, e_y, e_z$ ). Since the contact force at the arm's end-effector is always present, the Cartesian error components can be driven to zero only along the unconstrained directions. In fact, the plot shows that the error is almost entirely confined to the  $x$ -axis, settling at a steady-state value of approximately



**Figure 9:** Quadrotor Position Error Norm for centralized control for Case B (External Wrench). This plot shows the time evolution of the quadrotor's position error norm under a persistent external wrench applied to the end-effector with  $K_p = \text{diag}(10I_6, 100I_3)$ , with a damping ratio of 0.5 for the UAV components and 0.85 for the manipulator components. This configuration models a compliant behavior for the UAV and a rigid behavior for the manipulator.

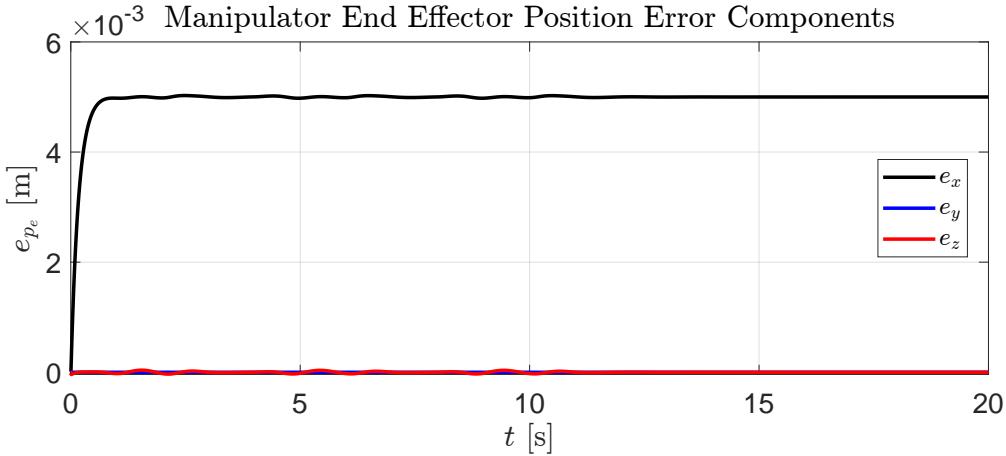


**Figure 10:** Quadrotor Orientation Error Norm for centralized control for Case B (External Wrench). This plot illustrates the time evolution of the quadrotor's orientation error norm for Case B, with  $K_p = \text{diag}(10I_6, 100I_3)$ , with a damping ratio of 0.5 for the UAV components and 0.85 for the manipulator components.

$5 \times 10^{-3}$  m due to the constant contact force of 0.5 N.

## 2.4 Conclusion

In conclusion, the Simulink implementation of the Cartesian impedance control for the UAV-manipulator system successfully replicated the behaviors of the case studies described in [1]. The results demonstrate the intended performance of each control strategy, with Case A<sub>1</sub> showing rigid low-error performance, Case A<sub>2</sub> showing a more compliant response with larger transient errors, and Case B revealing a stable but offset steady state due to the continuous external wrench. The analysis of the error norms provides evidence that the controller is robust and effective in managing different types of disturbances.



**Figure 11:** End-Effector Position Error components for centralized control for Case B (External Wrench). This plot displays the individual components end-effector position error for Case B, with  $K_p = \text{diag}(10I_6, 100I_3)$ , with a damping ratio of 0.5 for the UAV components and 0.85 for the manipulator components.

### 3 Decentralized Control of an UAM

The decentralized architecture separates the UAM into two subsystems, each with its own dedicated controller and trajectory planner. The UAV is managed by a hierarchical, passivity-based controller, inspired by the work of Ruggiero et al. [2]. To handle the dynamic disturbances induced by the manipulator’s movement, a momentum-based estimator is employed to calculate the external wrench and compensate for it in real-time. The manipulator, in turn, is governed by a Cartesian Impedance Controller. The performance of this decentralized scheme will be evaluated, and a comparative analysis will be drawn against a centralized control approach in 4.

The core assumption of this decentralized approach is that the UAV’s motion is quasi-static. This assumption allows the dynamic effects of the UAV on the manipulator to be neglected, thereby simplifying the control design. It is further justified by the fact that the manipulator’s controller typically operates at a faster rate, enabling it to effectively reject such disturbances. However, the manipulator’s movement creates a considerable disturbance on the UAV, which must be actively estimated and rejected.

#### 3.1 UAV Passivity-Based Controller

The UAV controller follows a hierarchical inner-outer loop structure based on the theory presented by Ruggiero et al [2].

##### 3.1.1 Momentum-Based Disturbance Estimator

It estimates the external wrench ( $f_{ext}, \tau_{ext}$ ) acting on the UAV by observing the system’s generalized momentum. The dynamics of the estimator are designed as a second-order filter to provide a smooth and accurate estimation of the wrench exerted by the manipulator’s motion. The output provides the estimated forces  $\hat{f}_e$  and torques  $\hat{\tau}_e$  used for feedforward disturbance cancellation.

##### 3.1.2 Outer Loop: Position Control

This loop generates the necessary commands to make the UAV track its desired position trajectory,  $p_{b,d}$ . A passivity-based control law computes a virtual acceleration command,  $\mu$ , based on the position error

$e_p = p_b - p_{b,d}$  and velocity error  $\dot{e}_p = \dot{p}_b - \dot{p}_{b,d}$ . The law is given:

$$\mu = \ddot{p}_d - (D_p \dot{e}_p + K_p e_p) \quad (11)$$

where  $K_p$  and  $D_p$  are diagonal positive-definite matrices of position and velocity gains, respectively.

### 3.1.3 Inner Loop: Attitude Control

The virtual acceleration command  $\mu$  is then translated into physical commands. A modified acceleration vector,  $\bar{\mu} = \mu - \frac{1}{m_b} \hat{f}_e$  is calculated by incorporating the estimated disturbance force. This vector is then used to compute the required total thrust  $u_T$  and the desired attitude angles ( $\phi_d, \theta_d, \psi_d$ ). The inner loop ensures the UAV tracks the desired attitude trajectory,  $\eta_d$ , generated by the outer loop. A passivity-based control law computes the body torques  $\tau_b^b$  required for stabilization. The control law for the inner loop:

$$\tau_b^b = Q(\eta_b)^{-T} (M(\eta_b) \ddot{\eta}_r + C(\eta_b, \dot{\eta}_b) - \hat{\tau}_e - D_o \nu_\eta - K_o e_\eta) \quad (12)$$

This equation incorporates the system's inertia (M) and Coriolis (C) matrices, the estimated disturbance torque  $\hat{\tau}_e$ , and feedback terms based on attitude errors.

## 3.2 Manipulator Cartesian Impedance Controller

The manipulator is controlled independently to follow its trajectory relative to the UAV's body. This controller calculates the joint torques required to make the end-effector follow its planned trajectory (`p_b_e_d`) while behaving like a desired spring-damper system. The control law implements Cartesian impedance control just like the one in the centralized section:

$$u = g(q) + J^T (B_x \ddot{x}_d + C_x \dot{x}_d + K_d \dot{\tilde{x}} + K_p \tilde{x}) \quad (13)$$

where:

- $u$  is the vector of joint torques.
- $g(q)$  is the gravity compensation term.
- $J$  is the manipulator Jacobian.
- $B_x$  and  $C_x$  are the inertia and Coriolis matrices in Cartesian space.
- $K_p$  and  $K_d$  are the desired stiffness and damping matrices.
- $\tilde{x}$  and  $\dot{\tilde{x}}$  are the position and velocity errors of the end-effector.

### 3.3 Wrench Transformation

To compute the dynamic effects of one system on the other, the force transformation formula between two frames of a rigid body has been used:

$$\begin{bmatrix} f_1^1 \\ \mu_1^1 \end{bmatrix} = \begin{bmatrix} R_2^1 & O \\ S(r_{12}^1)R_2^1 & R_2^1 \end{bmatrix} \begin{bmatrix} f_2^2 \\ \mu_2^2 \end{bmatrix}.$$

This formula has been specifically employed to calculate the effects of the manipulator on the UAV. As previously discussed, the dynamic effects of the UAV's motion on the manipulator are assumed to be negligible.

### 3.4 Simulations - Disturbance Rejection

This section presents the results obtained from the simulation of a decentralized control strategy for the UAV-manipulator system. In this approach, the objective is for the manipulator to remain still while the UAV maintains a stable, quasi-static hovering position. The decentralized control is designed to have the UAV's passivity-based controller and momentum-based estimator actively compensate for disturbances, including those from the manipulator. The exact same case studies from the centralized control analysis were used for a comparative evaluation. The simulation time for all cases was 20s. An external force, simulating a windy situation, was modeled as a sine wave with a frequency of  $\pi/2$  rad/s and an amplitude of 1 N. This force was applied along the  $x_b$  axis of the UAV's body frame ( $\Sigma_b$ ) for the first 10 seconds.

#### 3.4.1 Controller Gains

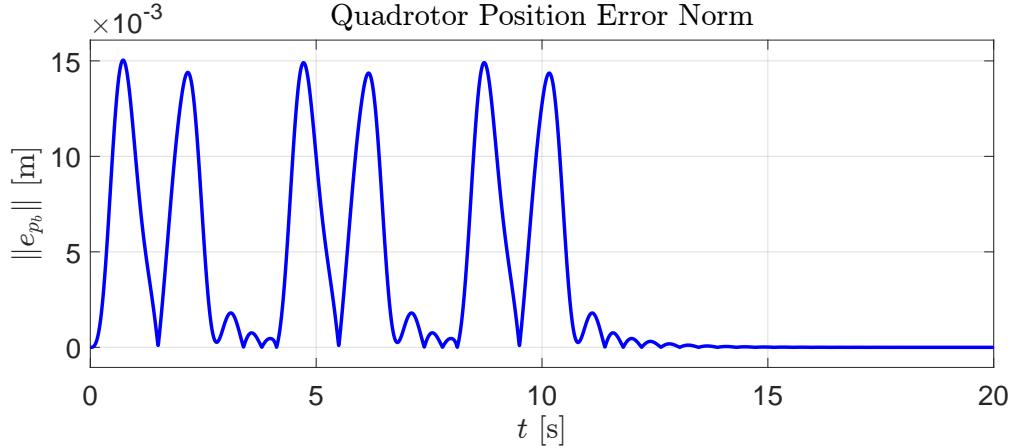
The controller gains for each simulation case were selected to obtain behaviors comparable to those observed in the centralized system. Different gains were applied to the UAV's outer and inner loops, as well as to the manipulator's impedance controller, to achieve the desired behaviors. In particular the inner loop was designed with higher gains than the outer loop, since the attitude control must react faster.

- **Case A<sub>1</sub> (Rigid Behavior):**
  - **UAV Controller:** The position loops was configured with a stiffness of  $K_p = 10I_3$  while the attitude loop was configured with a stiffness of  $K_p = 80I_3$  and both with a damping ratio tuned to 0.8.
  - **Manipulator Controller:** A stiffness of  $K_p = 80I_3$  was used to ensure that the manipulator remained rigidly in place.
- **Case A<sub>2</sub> (Compliant Behavior):**
  - **UAV Controller:** The stiffness gain of the position loop was kept at  $K_p = 10I_3$  but the gain of the attitude loop was put at  $K_p = 50I_3$ . The damping ratio was lowered to 0.5 to allow for a more compliant response to the external force.
  - **Manipulator Controller:** The same stiffness of  $K_p = 10I_3$  was used.
- **Case B (External Wrench):**
  - **UAV Controller:** The gains were kept at  $K_p = 10I_3$  and  $K_p = 50I_3$  with a damping ratio of 0.5.
  - **Manipulator Controller:** Configured with a stiffness of  $K_p = 100I_3$  and a damping ratio of 0.85 to maintain contact with the wall while allowing for compliance along the contact axis.

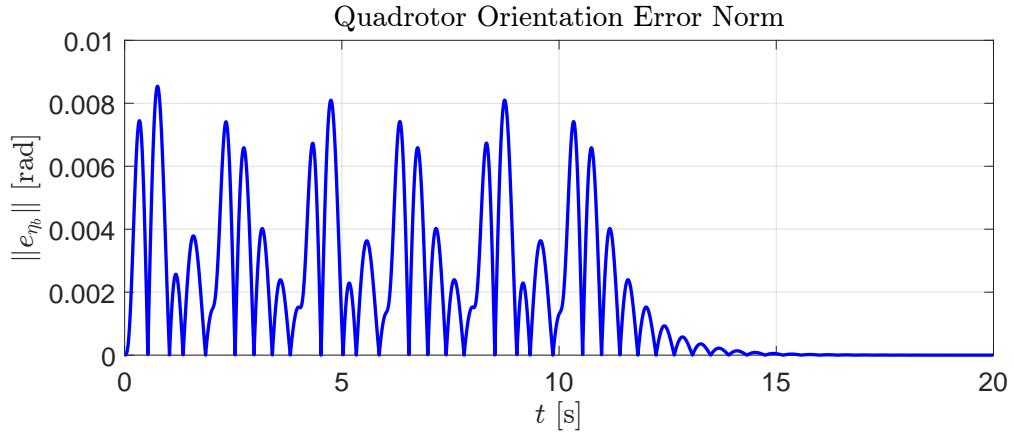
#### 3.4.2 Analysis of Case A: Rigid vs. Compliant Behavior

The results for Case A highlight the system's ability to maintain a hovering position under an external disturbance while the manipulator remains still. For both the rigid ( $A_1$ ) and compliant ( $A_2$ ) control settings, the manipulator's end-effector position error remains exactly zero throughout the simulation. This is a direct and expected outcome of the decentralized control architecture. Since the dynamic effects of the UAV on the manipulator are intentionally neglected in the controller design, the simulated manipulator is completely blind to the UAV's compensatory movements. Consequently, the controller does not register any disturbance, and the manipulator artificially maintains a perfect stationary position with zero error. It is critical to recognize that this perfect performance is a feature of the simplified model. The UAV's passivity-based controller, meanwhile, exhibits an oscillatory response to the external wind force. As seen in Figure 12, the quadrotor position error norm ( $\|e_{p_b}\|$ ) for Case  $A_1$  shows oscillations with a peak amplitude

of approximately 0.015 m for the first 10 seconds. After the force is removed, the error dampens and settles to zero. Similarly, the quadrotor orientation error norm ( $\|\eta_b\|$ ) in Figure 13 displays a peak amplitude of roughly  $8.5 \times 10^{-3}$  rad during the same period, before also converging to zero. For Case  $A_2$ , which models



**Figure 12:** Quadrotor Position Error Norm for decentralized control for Case  $A_1$  (Rigid Behavior). This plot shows the time evolution of the quadrotor's position error norm when the system is controlled with a rigid behavior ( $K_p = 10I_3$  for the position loop,  $K_p = 80I_3$  for the attitude loop and damping ratio tuned to 0.8) under external forces applied to the UAV.

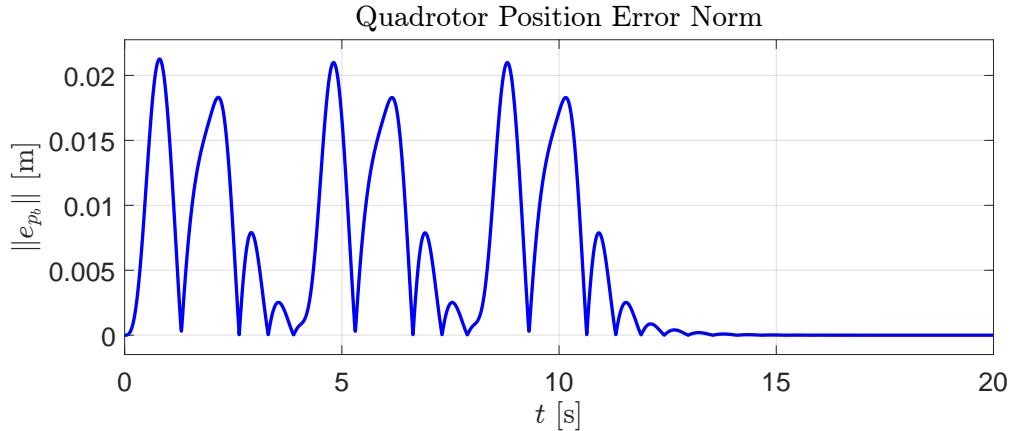


**Figure 13:** Quadrotor Orientation Error Norm for decentralized control for Case  $A_1$  (Rigid Behavior). This plot shows the time evolution of the quadrotor's orientation error norm under the rigid control strategy ( $K_p = 10I_3$  for the position loop,  $K_p = 80I_3$  for the attitude loop and damping ratio tuned to 0.8).

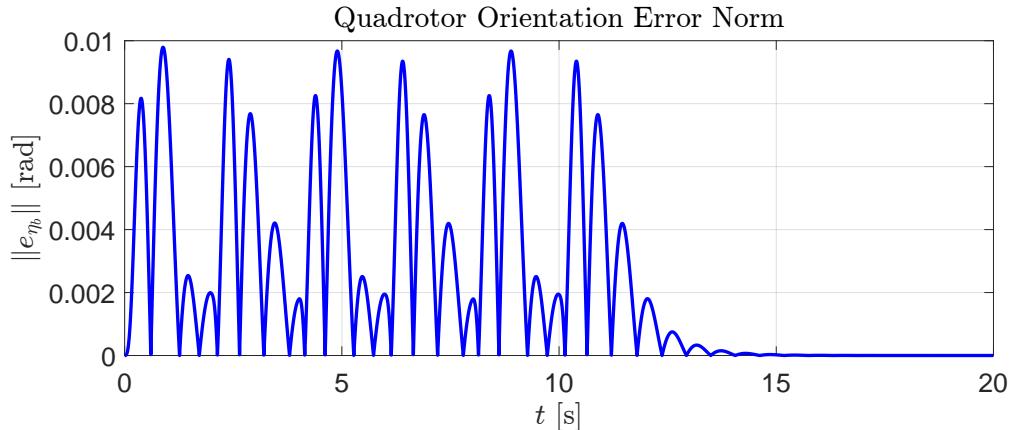
compliant behavior, the quadrotor position and orientation error norms (Figures 14 and 15) again show a oscillatory pattern for the first 10 seconds, with peak amplitudes of approximately 0.02 m and 0.01 rad, respectively. In both scenarios, the decentralized control successfully stabilizes the system after the external force is removed.

### 3.4.3 Analysis of Case B: External Wrench on Manipulator

Case  $B$  simulates both an external wind force on the UAV and a constant contact force of 0.5 N applied to the end-effector. The contact force representing interaction with a surface like a wall is therefore applied along the x-axis axis. Instead of analyzing the error norm Figure 16 plots the individual components ( $e_x, e_y, e_z$ ) of



**Figure 14:** Quadrotor Position Error Norm for decentralized control for Case A<sub>2</sub> (Compliant behavior). This plot shows the time evolution of the quadrotor's position error norm when the system is controlled with a compliant control behavior ( $K_p = 10I_3$  for the position loop,  $K_p = 50I_3$  for the attitude loop and damping ratio tuned to 0.5) under external forces applied to the UAV.

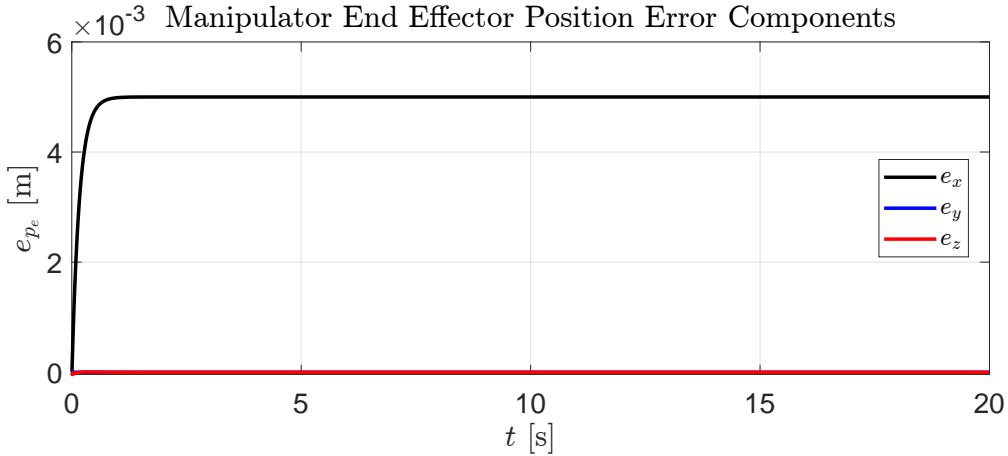


**Figure 15:** Quadrotor Orientation Error Norm for decentralized control for Case A<sub>2</sub> (Compliant behavior). This plot shows the time evolution of the quadrotor's orientation error norm under the compliant control strategy ( $K_p = 10I_3$  for the position loop,  $K_p = 50I_3$  for the attitude loop and damping ratio tuned to 0.5).

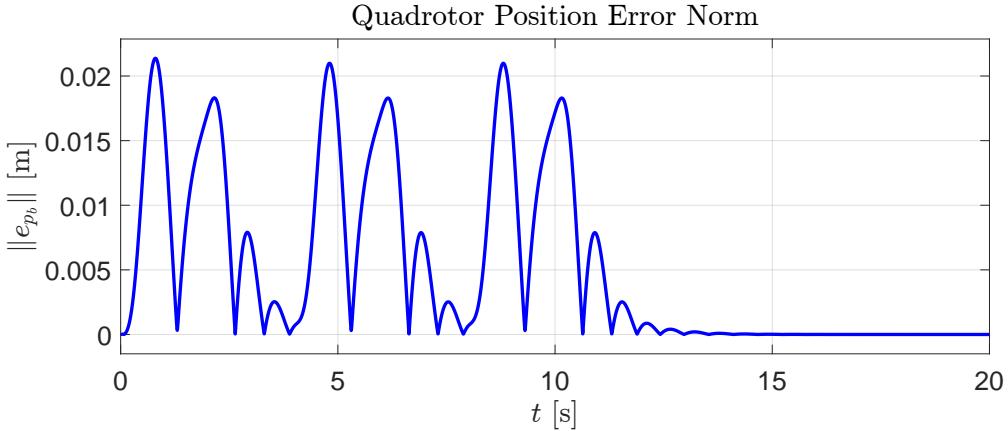
the end-effector position error. The plot clearly shows that the error is almost exclusively in the x-direction, settling at a steady-state value of approximately  $5 \times 10^{-3}$  m. The errors in the y and z directions remain zero. This is a deliberate and desirable outcome of the Cartesian impedance controller, it allows for a compliant displacement along the direction of the external force to maintain the contact. The quadrotor's position and orientation error norms (Figures 17 and 18) show a response similar to the Cases A . The quadrotor position error norm exhibits a peak of approximately 0.02 m and then converges to zero. The quadrotor orientation error norm has a peak of approximately 0.01 rad and then converges to zero. The oscillatory nature of the UAV's response for the first 10 seconds again corresponds to the period during which the windy force is active.

### 3.5 Conclusion

The simulation shows seemingly perfect manipulator control in Case A, with zero end-effector error, but this is a direct consequence of the model's simplification that decouples the two subsystems. While this highlights the ideal outcome of such a strategy, it also exposes that the simulation is blind to the physical



**Figure 16:** End-Effector Position Error for decentralized control for Case B (External Wrench). This plot displays the end-effector position error for Case B, with  $K_p = 100I_3$  and damping ratio tuned to 0.85, showing that the error is confined to the x-axis (direction of contact).



**Figure 17:** Quadrotor Position Error Norm for decentralized control for Case B (External Wrench). This plot shows the time evolution of the quadrotor's position error norm under a persistent external wrench applied to the base with  $K_p = 10I_3$  for the position loop,  $K_p = 50I_3$  for the attitude loop and damping ratio tuned to 0.5.

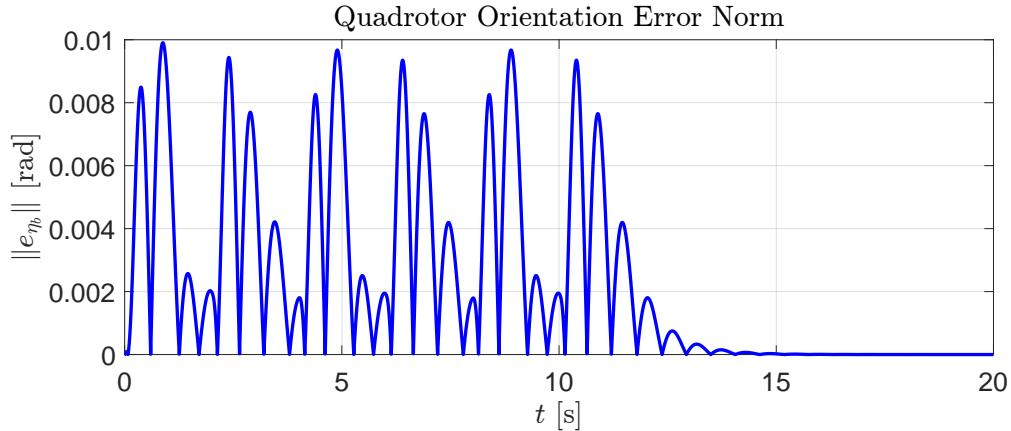
coupling that would exist in reality. In Case B's interaction scenario, the controller did successfully confine the position error to the direction of contact, demonstrating a correct impedance behavior. The momentum-based disturbance estimator proved capable of compensating for the external forces on the UAV.

## 4 Comparative Analysis of Control Strategies

This section provides a comparative analysis of the centralized and decentralized control architectures for the UAV-manipulator system, based on the simulation results from the three case studies. The objective is to evaluate the performance of each approach, particularly in how they manage external disturbances.

### 4.1 Performance

The primary difference between the two approaches lies in their control philosophy. The centralized strategy uses a fully coupled dynamic model. In contrast, the decentralized strategy relies on a reactive approach, where its subsystems handle disturbances independently.



**Figure 18:** Quadrotor Orientation Error Norm for decentralized control for Case B (External Wrench). This plot illustrates the time evolution of the quadrotor’s orientation error norm for Case B, with  $K_p = 10I_3$  for the position loop,  $K_p = 50I_3$  for the attitude loop and damping ratio tuned to 0.5.

A key element of the decentralized architecture is its momentum-based estimator. During the wind disturbance this estimator allows the UAV to actively predict and counteract the external force. But the most significant distinction emerges in the wall interaction scenario.

In the **centralized system**, the unified impedance control treats the constant force on the end-effector as a disturbance to the entire system. To manage this force, the controller allows both the manipulator and the UAV to shift their positions, resulting in a steady-state position error for the UAV. This is an intentional, globally compliant behavior where the whole system adapts to the contact force.

In the **decentralized system**, the response is different due to its decoupled nature. The manipulator’s controller handles the contact force locally, treating its base (the UAV) as fixed and therefore acting only on its joint positions. The UAV’s controller, on the other hand, perceives the dynamic effects transmitted from the manipulator merely as external disturbances to be rejected, without any constant force contribution from the wall. Its sole objective is to maintain a stable hovering position, which results in no steady-state position error for the UAV due to the wall contact.

## 4.2 Conclusion

The centralized control strategy provides a more unified and precise solution, resulting in smoother, faster convergence and minimal error norms. It is ideal for applications where low error and quick stabilization are central. The decentralized control excels at task isolation. The momentum-based disturbance estimator enables robust UAV positioning, particularly in maintaining a precise hover even when the manipulator is under a constant external load. The choice between architectures ultimately depends on the application’s priority.

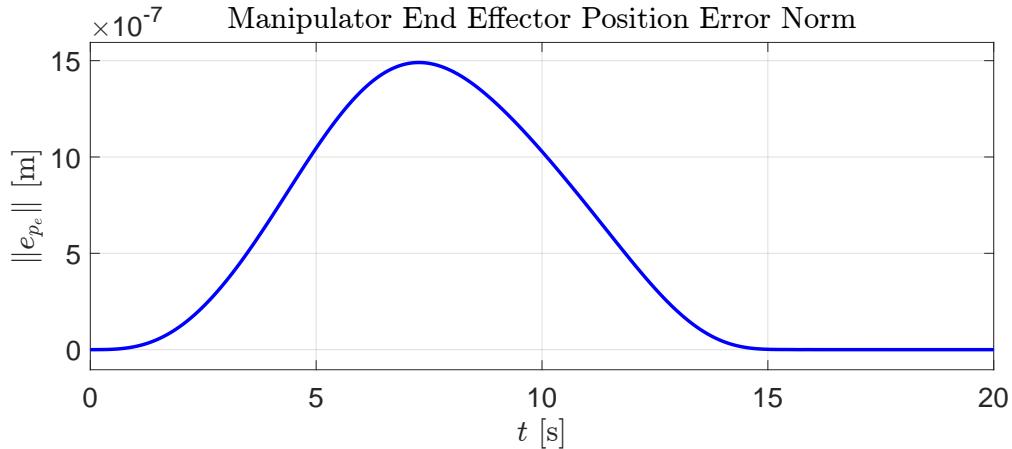
## 5 Comparative Analysis with moving Manipulator trajectory

This section presents a comparative analysis of the centralized and decentralized control strategies when the manipulator is commanded to execute a planned trajectory. The manipulator’s end-effector moves from its initial position to a final position of  $[0.1561; 0; -0.2061]$  relative to the UAV’s body frame, such that the final joint torque position will be  $q = [-\pi/2 + \pi/4; 0; \pi/2 - \pi/4]$ . The gains for this case were set to be the same as those used for Case A<sub>1</sub>, ensuring a direct comparison of the controllers’ performance in managing the internal dynamic disturbances. No external wind or contact forces were applied during this simulation. This

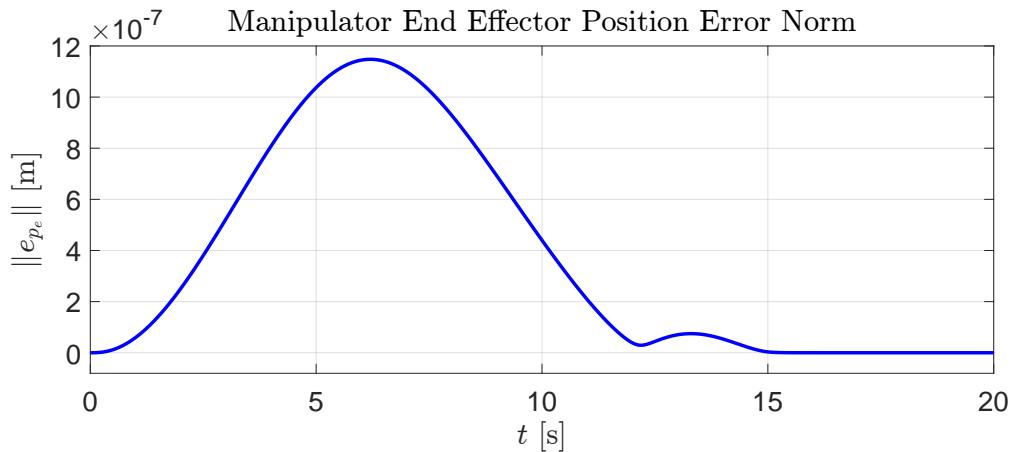
case is designed to evaluate how each control strategy manages the dynamic disturbances generated by the movement of the manipulator.

### 5.1 End-Effector Performance

In this scenario, both control strategies demonstrate good performance in the end-effector's ability to track the desired trajectory. As shown in Figure 19 for the centralized case and Figure 20 for the decentralized case, the end-effector position error norms are extremely small, on the order of  $1.5 \times 10^{-6}$  m and  $1.2 \times 10^{-6}$  m. This indicates that both controllers, regardless of their architectural differences, are effective in ensuring that the manipulator accurately follows its planned path.



**Figure 19:** End-Effector Position Error Norm for centralized control case. This plot displays the end-effector position error norm with  $K_p = 80I_9$  and damping ratio tuned to 0.8.

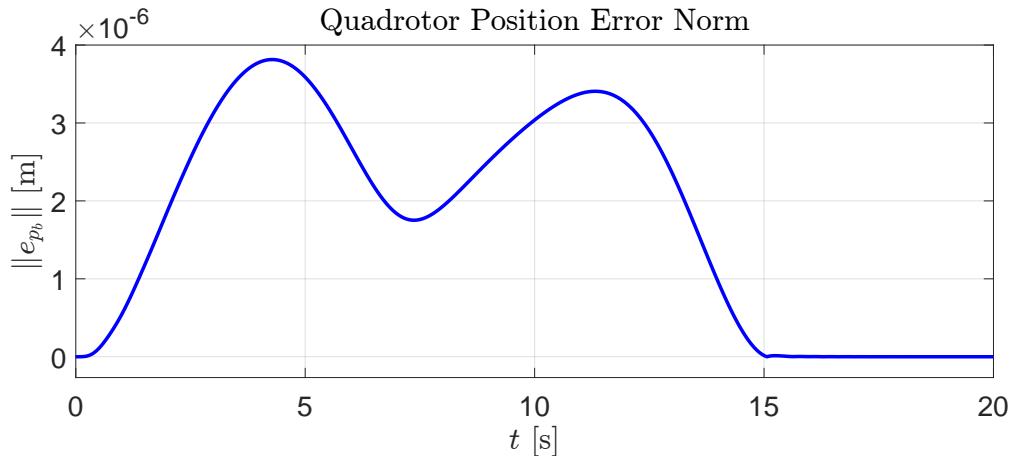


**Figure 20:** End-Effector Position Error Norm for decentralized control case. This plot displays the end-effector position error norm with  $K_p = 80I_3$  and damping ratio tuned to 0.8.

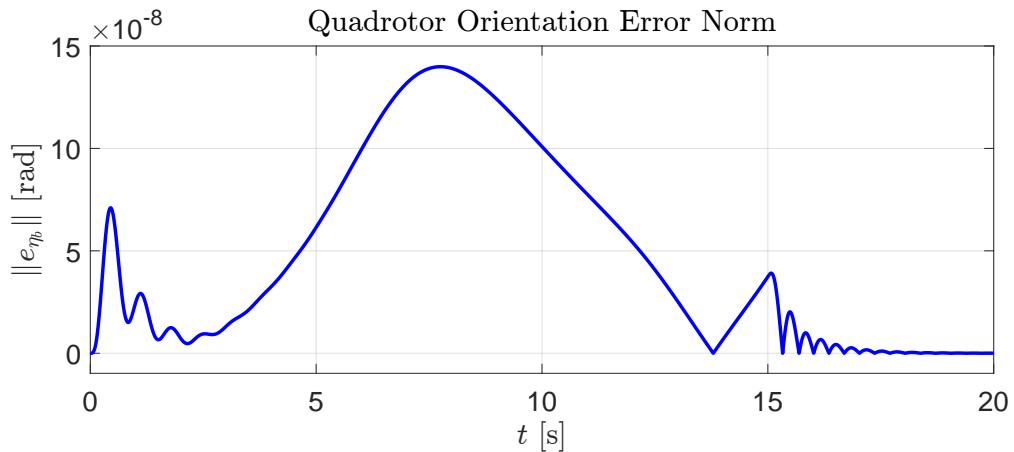
### 5.2 UAV Performance

The primary difference between the two control strategies is evident in the UAV's response to the manipulator's movement.

- **Centralized Control:** The centralized controller's response is smooth and precise. As the manipulator moves, the controller accounts for the dynamic coupling and commands the UAV to operate as expected. This results in an extremely low position error norm, which peaks at approximately  $4 \times 10^{-6}$  m (Figure 21), and an even lower orientation error norm, which peaks at approximately  $15 \times 10^{-8}$  rad (Figure 22).
- **Decentralized Control:** The decentralized approach, powered by its momentum-based estimator, proves to be equally effective. The estimator accurately predicts the wrench induced by the manipulator's motion and applies a compensatory action. This results in a stable hover, with the quadrotor position error norm peaking at only  $1.2 \times 10^{-6}$  m (Figure 23). The orientation error norm is also low, peaking at approximately  $1.4 \times 10^{-7}$  (Figure 24).



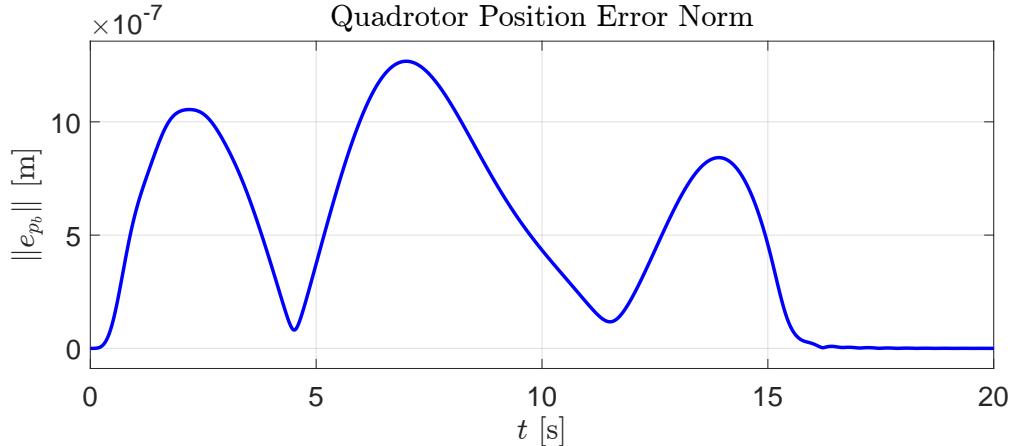
**Figure 21:** Quadrotor Position Error Norm for centralized control case. This plot displays the quadrotor position error norm with  $K_p = 80I_9$  and damping ratio tuned to 0.8.



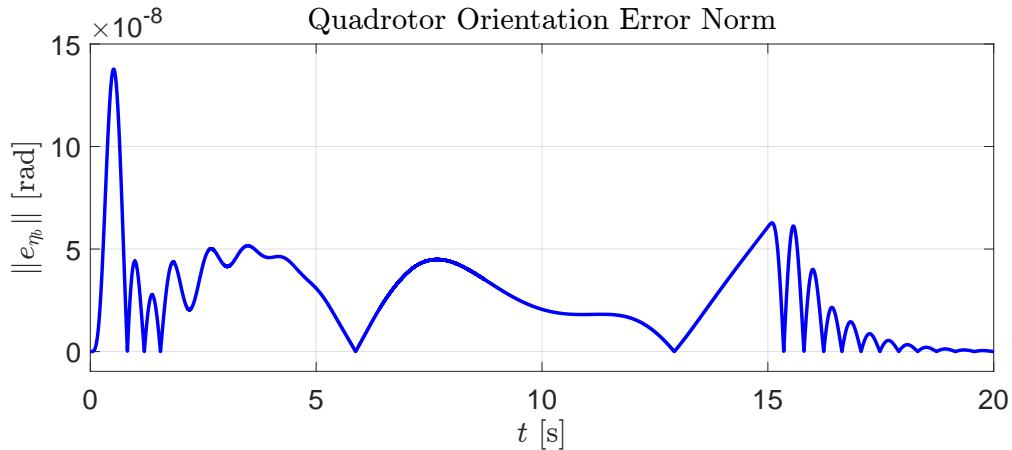
**Figure 22:** Quadrotor Orientation Error Norm for centralized control case. This plot displays the quadrotor orientation error norm with  $K_p = 80I_9$  and damping ratio tuned to 0.8.

### 5.3 Conclusion

The simulation of the moving manipulator case highlights the effectiveness of both control architectures in managing internal dynamic coupling. The centralized approach provides a superior, more coordinated



**Figure 23:** Quadrotor Position Error Norm for decentralized control case. This plot displays the quadrotor position error norm with  $K_p = 10I_3$  for the position loop,  $K_p = 80I_3$  for the attitude loop and damping ratio tuned to 0.8.



**Figure 24:** Quadrotor Orientation Error Norm for decentralized control case. This plot displays the quadrotor orientation error norm with  $K_p = 10I_3$  for the position loop,  $K_p = 80I_3$  for the attitude loop and damping ratio tuned to 0.8.

response, resulting in a smoother performance. Its comprehensive understanding of the UAM dynamics allows it to anticipate and negate disturbances more effectively. In contrast, the decentralized approach demonstrates that its disturbance estimator is highly capable of treating the manipulator's motion as a predictable disturbance and canceling it out effectively. The fact that both systems achieve end-effector accuracy confirms that both architectures are viable solutions for the UAM control problem.

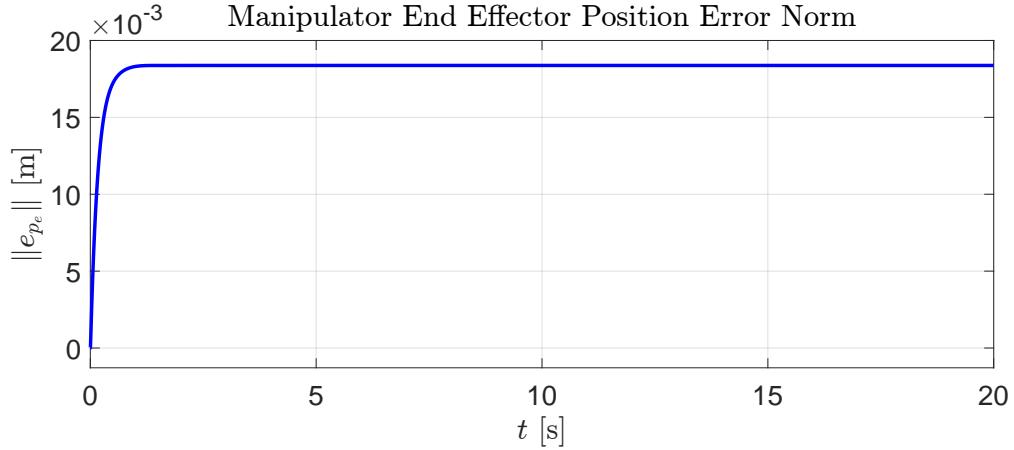
## 6 Comparative Analysis with moving Manipulator with a Payload

This case study evaluates the performance of both control strategies when a constant vertical force, simulating a 0.15 kg payload with a weight of 1.47 N, is applied to the manipulator's end-effector as it moves from its initial position to the same final position described in 5. The gains used for this case were the same as in case  $A_1$ .

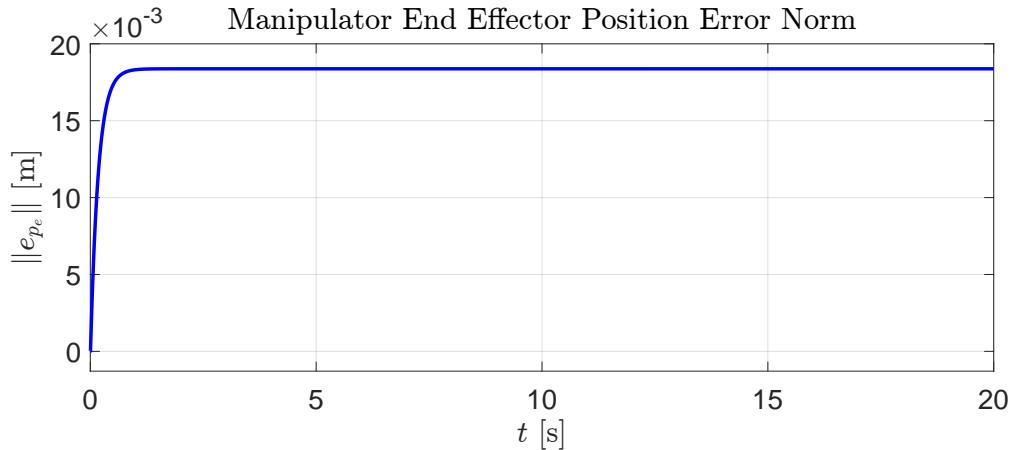
## 6.1 End-Effector and Quadrotor Position Error

For both control strategies, the end-effector and quadrotor position error norms exhibit a transient behavior during the manipulator's movement before settling to a non-zero steady-state value. As seen in Figures 25(centralized) and 26(decentralized) , the end-effector position error norm has an initial peak during the movement and then settles at a steady-state value of approximately  $1.8 \times 10^{-2}$ m in both cases. This final, non-zero error is the intended outcome of the impedance controller, which treats the payload's weight in the same way it treated the simulated wall in the previous sections.

The most significant difference lies in the quadrotor's position error norm (Figures 27 and 28 ).



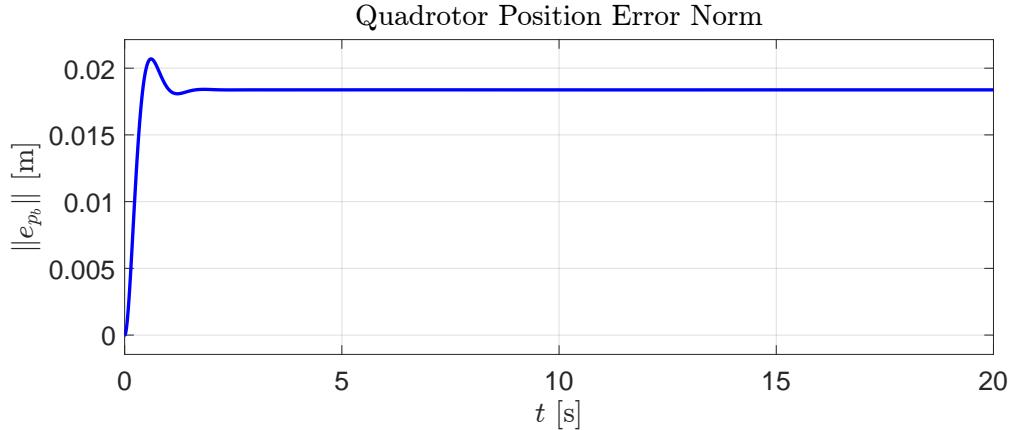
**Figure 25:** End-Effector Position Error Norm for centralized control case. This plot displays the end-effector position error norm with  $K_p = 80I_9$  and damping ratio tuned to 0.8.



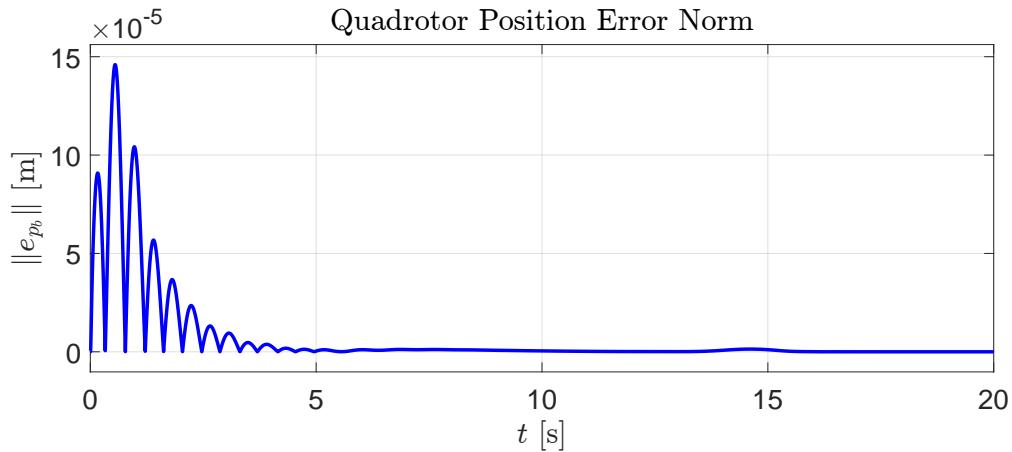
**Figure 26:** End-Effector Position Error Norm for decentralized control case. This plot displays the end-effector position error norm with  $K_p = 80I_3$  and damping ratio tuned to 0.8.

- **Centralized Controller:** The centralized controller exhibits a small overshoot in the quadrotor's position error norm (Figure 27) and a damped oscillatory response in its orientation error norm (Figure 29) before settling to a steady-state UAV position error of approximately 0.018 m as part of a global solution where both the UAV and the end-effector share the effort of counteracting the payload.
- **Decentralized Controller:** The performance in UAV positioning (a near-zero steady-state error of  $2 \times 10^{-5}$ m) is a direct consequence of the decoupling assumption. The UAV's controller has a singular

objective: to reject any and all wrenches at its base to maintain its hovering position.



**Figure 27:** Quadrotor Position Error Norm for centralized control case. This plot displays the quadrotor position error norm with  $K_p = 80I_9$  and damping ratio tuned to 0.8.

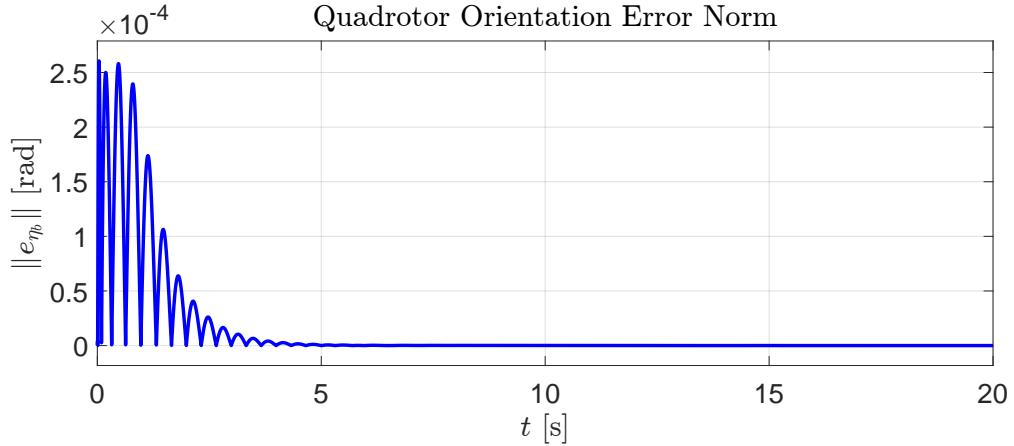


**Figure 28:** Quadrotor Position Error Norm for decentralized control case. This plot displays the quadrotor position error norm with  $K_p = 10I_3$  for the position loop,  $K_p = 80I_3$  for the attitude loop and damping ratio tuned to 0.8.

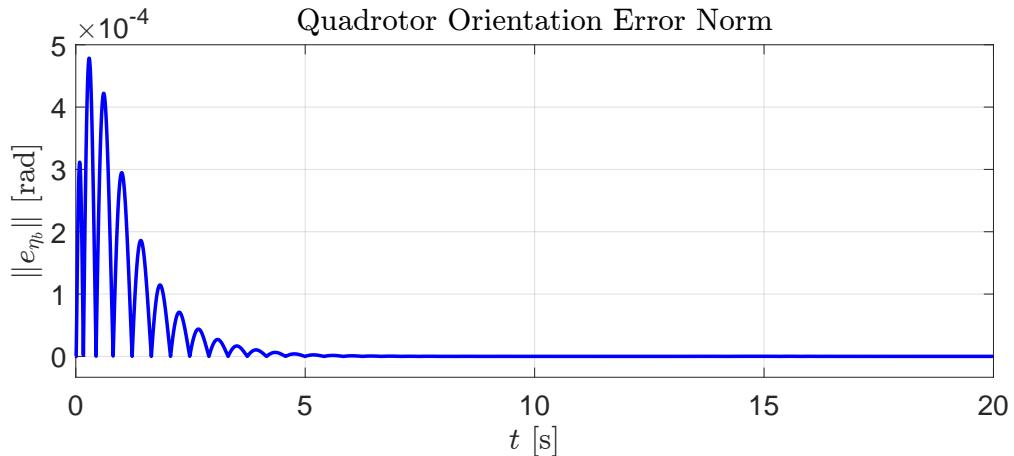
## 6.2 Conclusion

The payload simulation highlights the trade-offs of each control approach. The centralized controller must reduce hovering accuracy to control the entire system as a single, coupled entity. In contrast, the decentralized controller achieves better hovering performance due to its simplified structure.

By assuming the UAV's dynamics have a negligible effect on the manipulator, the system can be decoupled into two separate controllers. The manipulator controller handles only the local impedance task, while the UAV controller manages global positioning by actively compensating for all measured disturbances. This division of responsibilities allows the UAV to reach a much lower steady-state position error. This suggests that for tasks where maintaining the UAV's stability is the highest priority during payload interaction, a decentralized approach may offer several advantages.



**Figure 29:** Quadrotor Orientation Error Norm for centralized control case. This plot displays the quadrotor orientation error norm with  $K_p = 80I_9$  and damping ratio tuned to 0.8.



**Figure 30:** Quadrotor Orientation Error Norm for decentralized control case. This plot displays the quadrotor orientation error norm with  $K_p = 10I_3$  for the position loop,  $K_p = 80I_3$  for the attitude loop and damping ratio tuned to 0.8.

## References

- [1] Vincenzo Lippiello and Fabio Ruggiero. Cartesian impedance control of a uav with a robotic arm. *IFAC Proceedings Volumes*, 45(22):704–709, 2012.
- [2] Fabio Ruggiero, Jonathan Cacace, Hamid Sadeghian, and Vincenzo Lippiello. Passivity-based control of vtol uavs with a momentum-based estimator of external wrench and unmodeled dynamics. *Robotics and Autonomous Systems*, 72:139–151, 2015.