TECHNISCHE UNIVERSITÄT WIEN

AUTOMATION & CONTROL INSTITUTE
INSTITUT FÜR AUTOMATISIERUNGS- & REGELUNGSTECHNIK

Control of a Suspended Aerial Platform for Contact-based Applications

DIPLOMA THESIS

Conducted in partial fulfillment of the requirements for the degree of a

Diplom-Ingenieur (Dipl.-Ing.)

supervised by

Univ.Prof. Dipl.-Ing. Dr.-Ing. Christian Ott
Univ.Ass. Hemjyoti Das MSc

submitted at the

TU Wien
Faculty of Electrical Engineering and Information Technology
Automation and Control Institute

by
Sandra Martina Foith
Matriculation number 11808322

Vienna, February 2024

Robotic Systems Lab
A-1040 Wien, Gußhausstr. 27–29, Internet: https://www.acin.tuwien.ac.at

Preamble

This thesis was realized at the TU Wien, in particular at the Automation and Control Institute. I would like to express my gratitude to Univ. Prof. Dipl.-Ing. Dr.-Ing. Christian Ott, who was leading the project and Univ. Ass. Hemjyoti Das MSc, who supervised me with his expertise. From the very beginning, I was fascinated by the idea of developing a controller for a floating-base platform for contact-related applications. During this time, I was not only able to gain comprehensive knowledge but also valuable experience.

The extensive professional knowledge of Mr. Ott and his clear guidance were fundamental in achieving the realization of this project. My sincere appreciation also goes to Mr. Das for his continuous support and valuable guidance that helped me to implement the project in the best possible way.

With heartfelt thankfulness, I would like to express my gratitude to all the people who have made a valuable contribution to this project and who have accompanied and supported me along the way.

Vienna, February 2024

Abstract

This thesis is focused on the ongoing development in aerial robotic manipulation, which has made significant progress in recent years due to its diverse applications in areas such as inspection, maintenance, goods delivery and construction. The main purpose is to provide a simulated demonstration of precise multitask control of a suspended aerial platform with an attached robotic arm. The aerial platform (e.g. a multirotor or multicopter) is mounted from a crane and the focus is on achieving whole-body compliance control during aerial physical interactions with unknown environments.

Starting with an extensive literature review on aerial platforms, the thesis focuses on compliance control methods for aerial manipulators in physical interactions with their environment. Additionally, particular interest is given to the maximization of the force exerted from the endeffector, especially in situations involving contact with an unknown environment. For this purpose, a precise optimization problem is formulated and investigated in detail. All concepts are subjected to a detailed examination process to ensure their efficiency and effectiveness, which is reflected in the experiments.

The whole-body controller, implemented as a hierarchical controller, demonstrates the ability to perform several tasks simultaneously in various experiments. The hierarchy is reflected in the execution of the tasks by giving the highest priority to task 1, while the other tasks operate in the respective null-space in order to avoid influencing tasks with higher priority. The controller is able to control the tasks in such a way that they converge in all experiments. Furthermore, external influences such as forces and torques are compensated whilst the control system is still able to perform the desired tasks successfully. The optimizer is used to maximize the force at the endeffector in any dimension, along all directions or predefined directions. The limits of the optimizer have been explored experimentally to ensure efficient performance.

In terms of software framework, different modules have been implemented that enable an easy exchange of the robot arm. This allows effortless integration of a new robot arm into the simulation and ensures the efficient use of different robot models and parameters in simulation-based applications.

Summarizing, both the optimizer and the controller perform the desired trajectory of the robot tasks precisely and reliably, not only maximizing the forces at the endeffector, but also efficiently suppressing external forces. Thus, the whole-body control provides fast and robust performance while the optimizer maximizes the force at the endeffector.

Kurzzusammenfassung

Diese Arbeit beschäftigt sich mit der kontinuierlichen Weiterentwicklung der Steuerung von flugfähigen Robotern, die in den letzten Jahren aufgrund ihrer vielfältigen Anwendungen in Bereichen wie Inspektion, Wartung, Warenlieferung und Bauwesen erhebliche Fortschritte gemacht haben. Das Hauptziel ist die Bereitstellung einer simulativen Anwendung zur präzisen Multitasking-Steuerung einer schwebenden Luftplattform mit einem montierten Roboterarm. Die fliegende Plattform (z.B. eine Drohne) ist an einem Kran montiert, wobei der Hauptfokus darauf abzielt, eine Ganzkörper-Nachgiebigkeitssteuerung während physischer Interaktionen in der Luft in unbekannten Umgebungen zu erreichen.

Basierend auf einer umfangreichen Recherche über fliegende Plattformen, konzentriert sich die Arbeit auf Methoden der Nachgiebigkeitsregelung für fliegende Manipulatoren bei physischen Interaktionen mit ihrer Umgebung. Besonderes Interesse gilt dabei der Maximierung der vom Endeffektor ausgeübten Kraft, insbesondere in Situationen, in denen ein Kontakt mit einer unbekannten Umgebung besteht. Zu diesem Zweck wird ein präzises Optimierungsproblem formuliert und im Detail untersucht. Alle Konzepte werden einem detaillierten Prüfungsprozess unterzogen, um ihre Effizienz und Effektivität sicherzustellen. Die Ganzkörpersteuerung ist hierarchisch implementiert und belegt in verschiedenen Experimenten die Fähigkeit, mehrere Aufgaben gleichzeitig auszuführen. Die Hierarchie spiegelt sich in der Ausführung der Aufgaben wieder. Aufgabe 1 erhält die höchste Priorität und die anderen Aufgaben operieren im jeweiligen Nullraum, um eine Beeinflussung von Tasks mit höherer Priorität zu vermeiden.

Der Regler ist in der Lage, die Aufgaben so zu steuern, dass sie in allen Experimenten konvergieren. Darüber hinaus werden äußere Einflüsse wie Kräfte und Drehmomente kompensiert, während das Regelsystem weiterhin in der Lage ist, die gewünschten Aufgaben erfolgreich auszuführen. Der Optimierer wird eingesetzt, um die Kraft am Endeffektor beliebig zu maximieren. Die Limits des Optimierers werden experimentell erforscht, um eine effiziente Leistung zu gewährleisten.

In Bezug auf die Softwareentwicklung wird ein Framework implementiert, das einen einfachen Austausch des Roboterarms ermöglicht. Dies erlaubt die mühelose Integration eines neuen Roboterarms in die Simulation und gewährleistet die effiziente Nutzung unterschiedlicher Robotermodelle und Parameter in simulationsbasierten Anwendungen.

Zusammenfassend ist hervorzuheben, dass sowohl der Optimierer als auch der Regler die gewünschte Trajektorie der Aufgaben präzise und zuverlässig ausführen und dabei nicht nur die Kräfte am Endeffektor maximieren, sondern auch externe Kräfte effizient unterdrücken.

Contents

1 Introduction

In recent years, the use of aerial robots has experienced a rapid development and opens up a wide range of opportunities in a variety of areas. With applications including various sectors such as aerial photography, industrial inspection and many more [1–6], the technology presents its diversity and flexibility.

The utilization of aerial robots enables precise and efficient inspections of hard-to-reach or dangerous locations, significantly reducing human risks and costs, and thereby covers just some of the advantages of this technology.

The development of this advancement is progressing continuously. One exciting potential that has emerged is aerial manipulation - the ability of drones to grab, transport and manipulate objects. This capability significantly expands the application areas of drones, allowing them to be used in logistics e.g. storing and transporting goods [1], especially in places that are difficult to be accessed for conventional vehicles. In construction industry [2], utilization is found in the area of delivering material or assisting with construction projects. Also rescue missions or the delivery of medical supplies [3], the removal of material from hazardous environments [4] or the placement of sensors [5] to monitor the environment and detect potential hazards, are supported by aerial manipulation.

In such environments, drones with manipulation capabilities could be used to perform inspections, collect samples or even carry out repairs without exposing human workers to high risk [2].

In summary, aerial manipulation offers an important solution for areas that are difficult or even impossible for humans to reach, whether due to high radiation, inaccessible positions or other hazards, including nuclear facilities [6] or environments with potential radioactive hazards.

Consequently, this thesis concentrates on the control of an aerial manipulator. More specifically, the research project focuses on suspended aerial platforms [6], which recently have been proposed for unknown environments to execute contact-based applications.

The main focus of the system is to perform manipulation tasks, whereby the existing redundancy of the system enables additional tasks to be executed. The use of all degrees of freedom strives for the ability to perform several tasks simultaneously. This can be realized by means of a whole-body controller to manage given demands. In contact-based applications, an additional feature being pursued is to maximize the endeffector force exerted by the robot.

The developed methods are specifically designed for an aerial system with 10 Degrees of Freedom (DoF) and serial kinematics, connecting a 7 DoF robot with an aerial base of 3 DoF. The multiple tasks contain the precise control of the endeffector as well as the minimization of oscillations in the aerial base. A separate priority-based task accurately describes the robot's elbow to prevent possible singularities due to ambiguous trajectory execution. As soon as the robot comes into contact with an object in the unknown environment, an optimizer can be activated to maximize the force exerted while taking necessary constraints into account.

The performance of the controller and the optimizer are first evaluated separately and then demonstrated in a combined use-case scenario in which the entire system encounters an object and then pushes it.

Additionally, the revised framework provides a simplified method for performing simulations with different types of robots. The structure of the controller and optimizer is deliberately generic and allows external users to easily adapt and modify them.

1.1 Structure of this Work

The structure of this thesis is organized in six chapters. In Chapter 2, an in-depth insight into the current state of the art of the robotic system used is provided, accompanied by a comprehensive discussion of the state of the art in the area of suitable controllers. Based on this detailed characterization of various controllers, the hierarchical controller is selected based on a review of benefits and drawbacks and described in Chapter 3. The area of optimization problems and its theoretical aspects, which are essential for a comprehensive understanding, is covered in Chapter 4. The optimization process itself is defined within this section as well.

Chapter 5, which is the main part of this thesis, describes the robotic system used in more detail and provides a precise description of the implementation of the controller and the optimizer. In this chapter, both concepts are subjected to a detailed test and evaluation phase, first separately and then combined in a further use-case. The concluding chapter, Chapter 6, summarizes the findings of the thesis and provides an outlook for the future.

2 State Of The Art

This chapter is intended to provide an overview of the current state of the art in control strategies for aerial manipulators and the development of these systems. This is done by taking into account existing literature.

Therefore, the first part of this chapter will go into more detail about aerial manipulation in form of a research overview. The novel active platform, proposed in [6], will be used for later simulations, analyses and experiments and is referred as a Suspended Aerial Manipulator.

The second part of this chapter offers an overview of different control techniques for an aerial manipulator, with the primary focus on the whole-body controller. It should be noted that only methods for redundant robotic systems are considered, as this aligns with the scope of the work. Furthermore, it is important to clarify that the focus of the mentioned state of the art controllers only applies to single aerial manipulators, not multiple ones.

Based on the advantages and disadvantages of each technique, in the third section an explanation is given regarding the chosen controller for further simulations and experiments.

2.1 Aerial Manipulation

The concept of aerial manipulation describes a field of robotics that deals with the ability of autonomous or semi-autonomous aircraft to perform a variety of tasks. These are mainly used in areas that prove to be dangerous for humans such as performing tasks at high altitudes (e.g.: bridge inspection, high-voltage electric lines inspection, rotor blade repairing of wind turbines). [7]

In addition, aerial manipulation is also used in the areas of transportation such as delivery of packages, mapping, surveying and measuring of environments, for gripping, assembling and disassembling of mechanical parts and many more. [7, 8]

The execution is mostly carried out with an Unmanned Aerial Vehicle (UAV), which is according to [3] an aircraft that is navigating semiautonomously or autonomously. It should be mentioned that in this context, an UAV itself does not have any physical interaction with the environment. An extension towards applications that facilitate

physical contact will only become possible through the integration of a robotic arm into the UAV. This combination enables a direct manipulation of the environment. [9]

2.1.1 Research Overview

The early development of the UAV actively interacting with the environment, which is an aerial manipulator, composed of an aerial carrier (which could be e.g.: a helicopter, a crane...) with a robotic arm attached on it, starts in the year 2010. In [6], the integration of a robotic arm into an UAV is accomplished using a low-weight 2 or 3 DoF robot (depending on the task) in order to fulfill simple and straightforward tasks. Exemplary, in [10], a small robot with 3 DoF is used for aerial inspection and manipulation with the environment to detect and localize obstacles.

Due to the limited number of tasks that can be performed by a few DoF and the desire to exploit aerial manipulation in more detail, [9] installed a 7 DoF robot on the main carrier instead. The fully actuated and redundant robot integrated on the UAV, which in this case is a helicopter (Figure 2.1), provides benefits in performing complex tasks in hard-to-reach areas. The working range of the robot is limited in order to avoid self-collisions and to keep the robot in a safe working area. Weighing almost 10 kilograms, the robot requires a corresponding equipment of the helicopter, which leads to 3.7 meter diameter rotor blades. This limits the control of the required target objects.

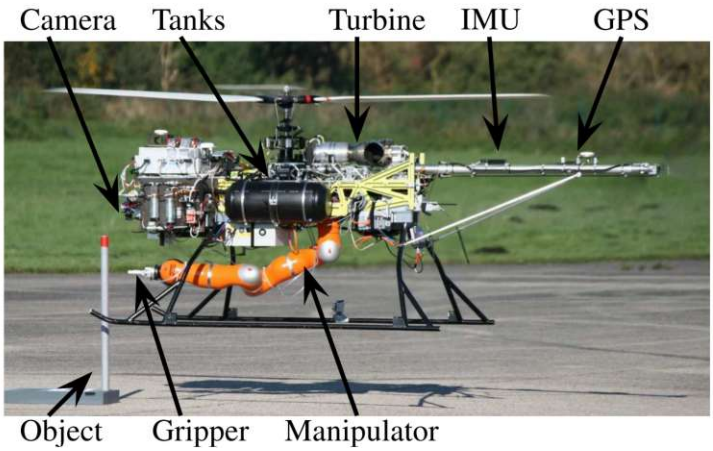


Figure 2.1: Setup for the aerial manipulation, reprinted from [9]

Y. S. Sarkisov et al. in [6] describes the subsequent prototype, which is designed in a way in order to overcome problems such as turbulences, ground effects, oscillations and the risk of collision between the rotor blades and the object of the environment. The design of the long reach manipulators includes a long flexible link between the main carrier and the UAV, i.e. the robot is no longer directly attached to the UAV. However, the flexible link results in the under-actuation of the system.

Another prototype with moving masses is proposed to dampen the oscillations caused by the long reach of the cable-suspended air manipulator. This prototype aims to improve the control of the system by adding additional masses that could counteract the vibrations and increase stability, but also results in additional load.

2.1.2 The Suspended Aerial Manipulator

The Suspended Aerial Manipulator (SAM) is an innovating development, emerged from the historical evolution, combining capabilities from both, robotic manipulator and UAV. [6]

The prototype of the SAM is a novel platform equipped with a 7-DoF robotic manipulator, that uses propulsion units and winches for control. Instead of attaching the robotic manipulator directly to an aerial carrier, the SAM is mounted on an active platform suspended by a cable, illustrated in Figure 2.2. The design enables a reduction in weight and dimensions of the propulsion units, increasing safety by avoiding the risk of collision between rotor blades and obstacles in the environment.

Figure 2.2: Prototype of Suspended Aerial Manipulator, reprinted from [6]

The platform is capable of generating independent forces and torques, enabling precise control of position and orientation. The SAM consists of several functional components, including a landing gear, a robotic manipulator, winches, and propulsion units (see Figure 2.2). The landing gear is designed to reduce weight and provides a large workspace for the manipulator. The robotic manipulator, a 7-DoF KUKA LWR 4, is mounted on the bottom side of the platform. The winches compensate for slow center of mass displacement during manipulation, while the propulsion units reduce dynamic deviations. This design allows for reduced energy consumption and increased safety.

In [6],the control strategy for the SAM includes modeling, preliminary control schemes and experimental results. The main carrier is assumed to be fixed during the manipulation task and the transportation of the SAM to the desired position is treated as a transportation problem with sling load. The robotic system is controlled using a cascade control scheme with control inputs for platform yaw-, platform position-, platform balance control and manipulator control. The SAM demonstrates successful response to control commands and external disturbances in experimental tests.

One further interesting development is presented in [11], in which Yigit et al. introduce a novel system configuration. In this setup, an aerial manipulator is suspended from a robotic carrier via a spring, as depicted in Figure 2.3. The system is controlled by a computed torque controller with proven stability, suitable for manipulation tasks. The spring compensates for gravity, allowing the robot for almost free-floating, which provides precise and stable manipulation. The system as a whole demonstrates enhanced energy efficiency, rapid dynamic response and a high accuracy level.

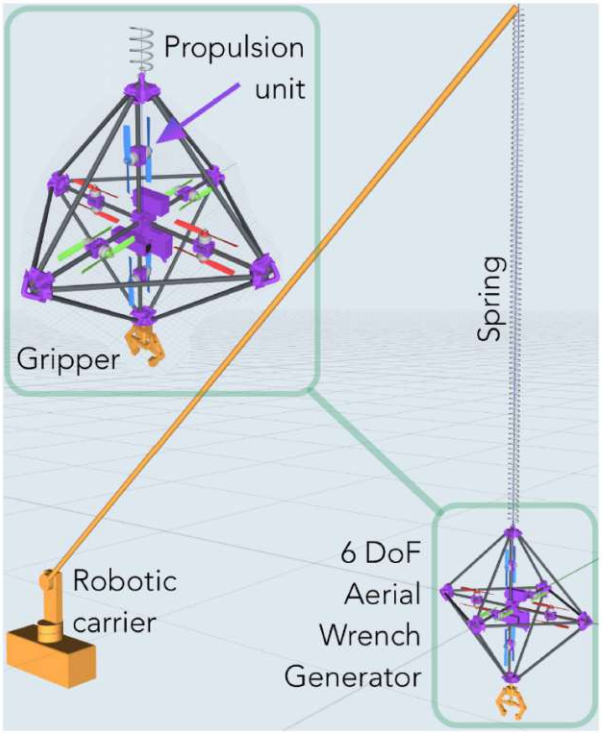


Figure 2.3: Structure the of Aerial Manipulator With Elastic Suspension, reprinted from [11]

2.2 State of the Art Controller

The previous Chapter 2.1.2 describes the control technique used for the operation with the SAM according to [6]. Further different control techniques and the required foundational knowledge are elaborated upon in this chapter.

2.2.1 Operational Space Formulation

The so-called Operational Space Formulation (OSF) is a task space- or operational space formulation first published in 1987 [12], that describes the desired behaviour of the robot in an intuitive form such as the endeffectors Cartesian space. The number of DoF to completely describe the position (translation and rotation) of the endeffector is called operational coordinates. The concept provides a way to define tasks and constraints in the task-space and to determine the corresponding joint torques or velocities required to accomplish these tasks. As per [13], most systems provide more actuated DoF than needed in task-space, the null-space is declared. With the OSF formulation the implementation is complex and requires high computational costs, furthermore overall stability is not provided, since "*...it is well known that this classical controller only gives a proof of stability for the main task, while the complete null space stability is unclear.*"[13] Moreover, [14] indicates that a thorough knowledge of the dynamic parameters is required for this purpose.

2.2.2 Cascade control strategy

As per [15], another approach for controlling an UAV is described with the Cascade Control Strategy (CCS), which involves a novel kinematic model derived using dual quaternion algebra and two controllers in a cascade scheme for tracking the desired trajectories and ensuring system stability and robustness to external disturbances.

Figure 2.4 illustrates the cascading structure of the controller, consisting of an inner- and outer loop. The inner loop consists of a linearization controller with partial feedback. The controller is designed to track the trajectory generated by the outer loop and regulate the controlled coordinates of the system while stabilizing the other generalized coordinates. The outer loop of the cascade control system is a kinematic controller. It is designed to perform trajectory tracking of the endeffector while keeping the entire system stable, even in the presence of uncertainties and external disturbances. The kinematic controller uses a linear H_∞ controller to calculate the motion references in the joint-space. These references are then tracked by the partial feedback linearization controller in the inner loop.

The results of the simulations in [15] show the effectiveness and robustness of the proposed control strategy.

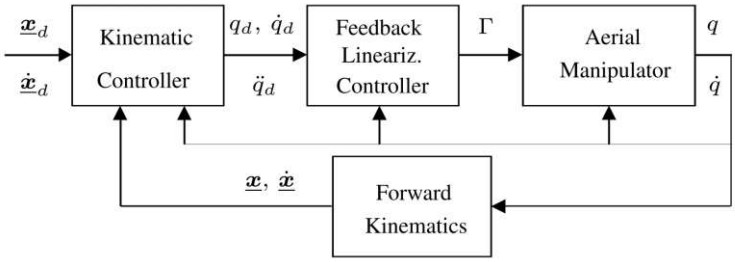


Figure 2.4: Control architecture of cascade control strategy, reprinted from [15]

2.2.3 Multi-task control

The following section focuses on two main control mechanisms, the impedance and compliance control, whereas both of them are used as the basis of common state of the art approaches, used for implementing multi-task controllers.

Impedance Control

Impedance-based control has become popular in recent years to control the robot in order to achieve its desired trajectory. Mostly, it is sufficient to control the position or the torque of the robot. However, when a robot is performing a task in which it interacts with the corresponding environment, exclusively commanding position- or torque control may not be sufficient. Therefore, when a task requires physical contact with the environment, impedance control is preferred, which will be discussed in more detail throughout this chapter. [16]

The control strategy considers the force exerted by the environment, whereby the environment can be physically described as an admittance according to [17], as it accepts force inputs and maps them to motion outputs (velocity). When the manipulator experiences contact from environment, the impedance control approach can modulate, regulate, and control the dynamic interaction between the manipulator and its environment by changing the impedance. Impedance control is thus an approach to achieve dynamic control of both, force and position by its relation.

One application example mentioned in [13] is the interaction between humans and robots, e.g. in households or when collaborating in the same workspace.

The definition of impedance In [16], Park describes a robotic system to be a dynamic system with changing energy over time. During physical contact with the environment, energy is exchanged between the two parties, defined by the multiplication of effort times flow. Exemplary in mechanical systems, effort equals force/torque and flow is velocity whereas in electrical systems, effort is electrical potential and flow represents current.

This leads to the resulting mechanical resp. electrical impedance. Whereas mechanical

impedance I_m is defined as output force (effort) F divided by input motion (flow) v, resulting in $I_m = F/v$, its analogous electrical impedance I_{el} is defined by the ratio of output voltage (effort) U and input current (flow) I, leading to $I_{el} = U/I$.

Control law The main goal of the impedance control in [13, 16, 17] is to adjust the mechanical impedance of the robot in order to control the dynamic interaction with the environment (admittance). By modulating and regulating the impedance, the dynamic interaction between the manipulator and the environment can be controlled and coordinated effectively. The desired impedance can be expressed by

$$M_d(\ddot{x} - \ddot{x}_d) + B_d(\dot{x} - \dot{x}_d) + K_d(x - x_d) = F_e, \tag{2.1}$$

in Cartesian coordinates. [16]

In (2.1), $x, \dot{x}$ and $\ddot{x} \in \mathbb{R}^n$ describe the position, the velocity and the acceleration of the endeffector and the corresponding desired position, speed and acceleration is represented by $x_d, \dot{x}_d$ and $\ddot{x}_d \in \mathbb{R}^n$. The acting force on the endeffector, emanating from the environment is defined as $F_e \in \mathbb{R}^n$. The positive definite matrices M_d, B_d and $K_d \in \mathbb{R}^{n \times n}$ describe the mass, damping and stiffness coefficients. These vary depending on the desired mechanical impedance.

N. Hogan in [17] indicates the dynamic relations within the manipulator variables in the control law: its stiffness defines the force output with the resulting displacement (extension or compression) of the spring. Equally, the damping coefficient affects the output of the force for a given velocity input. Given [14], the resulting desired contact behavior leads to precise and stable trajectory tracking while maintaining physical interactions with the environment.

Compliance Control

As described in [13, 16], in a special case of impedance control, called compliance control, the natural inertia of the robot is preserved. This means that the focus is put on achieving a desired contact stiffness and damping while maintaining the natural inertia of the robot. In other words, the robot responds to external forces and torques in a way that preserves the natural inertia of the system.

2.2.4 Hierarchical Controllers

The Hierarchical Controller (HC) [14, 18] is a control strategy used in robotics for managing multiple tasks in a systematic and organized manner, if the provided system is kinematically redundant. Redundancy implies that more DoF are provided than needed for executing the desired task. The controller allows for the utilization of this unused DoF and therefore allows for executing multiple tasks simultaneously by using a task hierarchy and prioritized execution.

The tasks are defined in a control task hierarchy with different levels of priority, the one with the most priority starting with 1. By using null-space projections, which dynamically decouple and prioritize tasks, the system can ensure that higher priority tasks are achieved while still considering lower priority ones. This approach enables the robot to perform complex tasks efficiently without compromising safety or stability.

The HC can be used for e.g.: trajectory tracking, compliance control and physical interaction with the environment.

Null-space projection

According to [14, 18], a null-space projector functions as a mathematical tool used in control systems to implement a strict hierarchy of control tasks according to their priority order. By projecting lower priority tasks into the null-space of the higher priority tasks, joint velocities associated with the lower priorities are adjusted to avoid interfering with the higher priority tasks. This leads to dynamic consistency of the null-space projection and ensures that subordinate tasks of lower priority do not interfere with tasks of higher priority.

2.2.5 Hybrid Adaptive Control

In [19], a Hybrid Adaptive Control (HAC) is introduced, which is a control scheme for achieving dynamic stability in a mobile manipulation UAV that uses a combination of gain scheduling and Lyapunov-based adaptive model reference control. The controller is designed to adapt to changes in the dynamics of the aerial manipulator and to provide stability during flight and manipulation tasks.

The gain scheduling adjusts the parameters of the controller based on the variability in the position of the manipulator joints and thus allows for adaptation to changes in the dynamics of the system. The adaptive model reference control continuously minimizes errors between system output and its reference model and stabilizes the system, which is proven to be stable using Lyapunov theory.

This hybrid approach, i.e., the combination of gain scheduling and adaptive model reference control thus provides a superior and robust control solution as well as stability during aerial manipulation missions.

2.2.6 Aerial manipulation using Reinforcement Learning

In accordance to [20], utilizing an intelligent planning method for aerial manipulation, such as Reinforcement Learning (RL), allows for the increase of the degree of intelligence of a robotic system.

The proposed method for the UAV in Figure 2.5 is composed of two main elements, which are the policy and the Markov decision process. The policy (more precisely the proximal

policy optimization) serves to learn the optimal policy for the agent, i.e. the robotic system. The Markov decision process is made of a set of states, actions, rewards, and a transition function. The agent interacts with the environment and receives a reward for each action. The action with the greatest reward is then included in the policy.

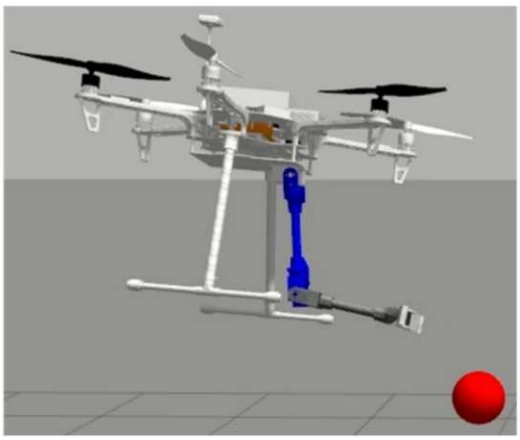


Figure 2.5: Aerial manipulation robot, reprinted from [20]

Therefore, [20] shows that RL is able to estimate the dynamic characteristics of the robotic system and thus enhances its robustness and is less complex than the process of dynamic analysis and modeling.

2.3 Comparative Analysis of State of the Art Techniques

Based on the summarized data from papers [12–20], Table 2.1 provides an overview of the controller techniques described in Chapter 2.2 and shows that the HC offers the most advantages and is therefore also used for further simulations and experiments of this thesis.

	Advantages	Disadvantages
OSF [12–14]	• Simplicity: in terms of task description	• Complexity: due to OSF formulation implementation • High computational costs • Overall stability is not provided: only of main task • A priori knowledge: of dynamic parameters required in order to control
CCS [15]	• Stability ensuring: even in presence of uncertainties • Control of underactuated systems: due to complex control strategy	• Complexity: due the occurrence of multiple control loops and their coordination with each other • Sensitivity to model uncertainties: since inaccurate models of system dynamics can compromise the effectiveness of cascade control, diminishing its practical robustness • Tuning challenge: due to multiple controllers in different loops • Computational costs: due to implemented loops, this also leads to problems in real-time applications where fast and efficient control is necessary • Limited Flexibility: due to specific control strategy

	Advantages	Disadvantages
HAC [19]	• Improved Flight Stability with gain scheduling • Enhanced Safety: by monitoring the position of the manipulator joints and adapting controller parameters accordingly • Expanded Use of UAVs: since adaptive control allows for interaction with environment • Improved Stability and Feasibility: with Lyapunov-based model reference adaptive control • Flexibility in Parameter Adaptation: gain scheduling allows for adaption of controller parameters	• Complexity: due to combination of different techniques • Tuning challenge: due to combination of different techniques, requires expertise • Limited Application Scope: since control scheme is specifically designed for mobile manipulator UAVs and not directly applicable to other types of aerial or ground-based robots • Computational Requirements: due to mandatory simulations and calculations to adjust the control loop gain and stabilize the system • Advantage = Disadvantage since parameter variations can lead so instability or sub-optimal performance
RL [20]	• Real-time planning and replanning: allowing the robot to adapt to changing environments and make dynamic decisions • Consideration of nonlinear interaction and strong coupling: ensuring that the agent pays more attention to the overall stability of the robot, leading to more efficient and effective manipulation • Improved disturbance degree to the robot: resulting in smoother and more accurate movements	• Complexity of the reward function: function is composed of multiple sub-parts, including penalties for time consumption, energy consumption, and leaving the designated work area • Resource consumption: leads to costly computational resources

	Advantages	Disadvantages
HC [14, 18]	• Simultaneous Execution of Subtasks: enables the robot to perform complex tasks efficiently • Preservation of Natural Inertia: of the robot ensures that the contact stiffness is accurately implemented and avoids distortion of external forces • Proposed technique eliminates the need for feedback of external forces • Beneficial Interaction Properties: combination of benefits of an impedance controller with the tracking capabilities of a dedicated tracking controller	• Modeling Errors and Uncertainties: modeling errors in the inertia matrix can deteriorate the control performance

Table 2.1: Listed advantages and disadvantages of each control technique

3 Hierarchical impedance-based Whole-Body Controller

As already derived in Section 2.2.4, a Hierarchical impedance-based Whole-Body Controller is defined as a control strategy, used in applications where the robot interacts with the environment. The robot may fulfill multiple task according to a defined hierarchical manner, while interacting with the environment. When working with the environment, the natural inertia of the system is preserved.

In order to fulfill a strict control task hierarchy, a dynamically consistent null-space projector is needed, leading to decoupled task-space velocities. The decoupled task-space coordinates can be used within the control law, since it guarantees that lower tasks do not disturb the higher tasks. Although the regulation in the decoupled space is sufficient, an inverse transformation into the original task-space is carried out, for more intuitiveness. The decoupled task-space coordinates have no resemblance to the original ones and can therefore lead to comprehension problems when reading out the results. Figure 3.1 shows the schematic and superficial structure of the controller.

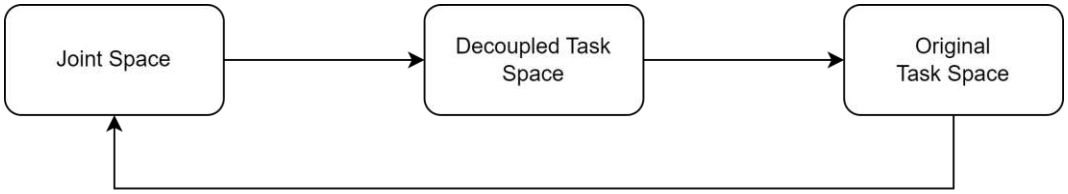


Figure 3.1: Proposed Controller Structure

This chapter deals with the construction of a hierarchical impedance-based whole body controller and describes all necessary parts. Thereby the required basics are repeated, different types of null-space projections are discussed and the transformations between the different spaces. At this point it should be mentioned, that this chapter discusses the controller operating in task-space.

Furthermore it is important to remark that, starting with Section 3.3.2, [14] serves as a basis unless otherwise stated.

3.1 Basics

According to Dietrich et al. [13], the Lagrangian formalism based Equations of Motion (EoM) for a robotic system with n DoF are described by

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau + \tau_{ext}, \tag{3.1}$$

denoting $q, \dot{q}$ and $\ddot{q} \in \mathbb{R}^n$ as joint configuration, it´s corresponding velocity and acceleration. $M(q) \in \mathbb{R}^{n\times n}$ is the positive definite and symmetric inertia matrix and $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ describes the Coriolis- and centrifugal terms of the system. The forces generated by gravity are summarized in $g(q) \in \mathbb{R}^n$, the control input, given as generalized forces is defined as $\tau \in \mathbb{R}^n$ and $\tau_{ext} \in \mathbb{R}^n$, represents external generalized forces.

By using the forward kinematics, the generalized coordinates in joint-space are mapped to the coordinates in task-space

$$x_i = f_i(q), \tag{3.2}$$

where x_i denotes the coordinates in task-space, defined for $i = 1...r$ task-space coordinates, and r defines the number of tasks fulfilled with the robotic system. The mapping from joint- to task-space is done with forward kinematics $f_i(q) \in \mathbb{R}^{m_i}$ corresponding the required task dimension $m_i \in \mathbb{N}$ to fulfill task i.

The appropriate task-space velocities are defined by the derivative of (3.2),

$$\dot{x_i} = \underbrace{\frac{\partial}{\partial q} f_i(q)}_{J_i(q)} \dot{q}, \tag{3.3}$$

$$\dot{x_i} = J_i(q)\dot{q}, \tag{3.4}$$

with the corresponding Jacobian matrices $J_i(q) \in \mathbb{R}^{m_i\times n}$ for all hierarchy levels $i = 1...r$, describing the mapping between joint- and task-space velocities.

3.2 Different types of null-space projection

Referring to Chapter 2.2.4 and [13, 14], the hierarchical priority based execution of tasks is fulfilled by null-space projection. The main priority task with priority $i = 1$ and $m_i < n$ DoF needed, is carried out by fully utilizing the capabilities of the robotic system. The second most important task then operates in the null-space of task 1 without disturbing it. This characteristic continues until task $i = r$.

Furthermore, it is assumed that the Jacobian matrices described below are non-singular, i.e. invertible.

Since null-space projections can be defined in different ways, successive and augmented projections are considered within this thesis.

In Dietrich et al. [13], both null-space projections address the initial scenario in which the task with the highest priority is executed equally, whereby all available capacities of the robot system are used for its completion. This corresponds to (3.4) for $i = 1$. Beginning from the second most important task, the projection into the null-space of the first priority task is performed. Accordingly, the null-space projector for task 1 results in the identity matrix I.

3.2.1 Successive Projections

In [13, 14], the projection of the second priority level is performed with the successive null-space projector $N_2^{suc}(q)$, evaluated by

$$N_2^{suc}(q) = I - J_1(q)^T (J_1(q)^{\#})^T, \tag{3.5}$$

determining $()^{\#}$ as pseudoinverse. The resulting decoupled task-space velocity v_2 does not interfere with the main task and is defined as

$$v_2 = J_2(q) N_2^{suc}(q) \dot{q}, \tag{3.6}$$

accordingly. The generalization of the projection with the successive formula leads to

$$N_i^{suc}(q) = N_{i-1}^{suc}(q) \Big(I - J_{i-1}(q)^T (J_{i-1}(q)^{\#})^T \Big) \tag{3.7}$$

and

$$v_i = J_i(q) N_i^{suc}(q) \dot{q}. \tag{3.8}$$

3.2.2 Augmented Projections

According to [13, 14], the null-space projector of the augmented projection is defined as

$$N_i^{aug}(q) = I - J_{i-1}^{aug}(q)^T (J_{i-1}^{aug}(q)^{\#})^T, \tag{3.9}$$

differing in definition from (3.7) in the Jacobian matrix used. The so-called augmented Jacobian matrix stacks all level-specific Jacobian matrices according to their task hierarchy down to level i and therefore contains all higher priority Jacobian matrices for a given i, see its definition below.

$$J_i^{aug}(q) = \begin{pmatrix} J_1(q) \\ \vdots \\ J_i(q) \end{pmatrix} \tag{3.10}$$

By the fact that the null-space projector for, e.g., priority level 3 with augmented projection using

$$N_3^{aug}(q) = I - J_2^{aug}(q)^T (J_2^{aug}(q)^{\#})^T = I - \begin{pmatrix} J_1(q) \\ J_2(q) \end{pmatrix}^T \left(\begin{pmatrix} J_1(q) \\ J_2(q) \end{pmatrix}^{\#} \right)^T \tag{3.11}$$

differs from the successive projector using

$$N_3^{suc}(q) = N_2^{suc}(q)\Big(I - J_2(q)^T (J_2(q)^{\#})^T\Big), \tag{3.12}$$

highlights its complexity, which results from the large number of rows in the augmented matrix and the resulting complex pseudo-inversion process.

Analogously, it is possible to define the augmented task-space velocities and accelerations. This is done by stacking the coordinates according to the hierarchy level $i = 1...r$

$$\dot{x_i}^{aug} = \begin{pmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_i \end{pmatrix} \tag{3.13}$$

and

$$\ddot{x}_i^{aug} = \begin{pmatrix} \ddot{x}_1 \\ \vdots \\ \ddot{x}_i \end{pmatrix}. \tag{3.14}$$

Similar to (3.4), the relation between augmented task-space velocities and joint velocities can be calculated using the augmented Jacobian matrix (3.10)

$$\dot{x}_i^{aug} = J_i^{aug}(q)\dot{q}. \tag{3.15}$$

Despite the fact that the augmented projector involves more computational effort, it is preferred. In [21], it is demonstrated that when successive projecting a lower priority task onto the null-space of the higher priority task, it cannot be guaranteed that the resulting mass matrix M in task-space will turn out to be a lower block-triangular matrix with positive-definite sub-matrices on the diagonal. In further consequence, stability conclusions cannot be established. The projection with an augmented projector, however, guarantees that M is always lower block-triangular matrix, as well as its sub-matrices on the diagonal are positive definite.

3.3 Decoupled task-space

The null-space projector according to [13, 14] is denoted as $N_i(q) \in \mathbb{R}^{n \times n}$ for $i = 1...r$, where r defines the amount of tasks and n represents the number of DoF of the system. $N_i(q)$ is defined differently for each level i of the hierarchy, as already mentioned in Section 3.2. This section also justifies why the augmented null-space projector is used. Its definition, referred to (3.9), is as follows:

$$N_i(q) = \begin{cases} I, & \text{for i = 1} \\ I - J_{i-1}^{aug}(q)^T (J_{i-1}^{aug}(q)^{\#})^T & \text{for i = 2...r} \end{cases} \tag{3.16}$$

with I defined as identity matrix, $J_{i-1}^{aug}(q)$ defined as the augmented Jacobian matrix for level $i-1$, $()^{\#}$ presents the dynamically consistent pseudoinverse and $()^T$ denotes the transpose operation.

3.3.1 Decoupled task-space coordinates

Expression (3.4) already indicates the mapping of the generalized coordinates from the joint-space into the task-space. By using the null-space projector (3.16), now the decoupled task-space velocities $v_i \in \mathbb{R}^{m_i}$ for $i = 1...r$ can be defined. Dietrich et al. in [14] thus formulates

$$\underbrace{\begin{pmatrix} v_1 \\ \vdots \\ v_r \end{pmatrix}}_{v} = \begin{pmatrix} J_1(q)N_1(q)^T \\ \vdots \\ J_r(q)N_r(q)^T \end{pmatrix} \dot{q}. \tag{3.17}$$

Since the first term of the above equation can be simplified (according to (3.16)), the term definition of v results in

$$\begin{pmatrix} v_1 \\ \vdots \\ v_r \end{pmatrix} = \underbrace{\begin{pmatrix} J_1(q) \\ J_2(q)N_2(q)^T \\ \vdots \\ J_r(q)N_r(q)^T \end{pmatrix}}_{\overline{J}(q)} \dot{q}. \tag{3.18}$$

The matrix multiplication of the Jacobian matrix and the associated projector is summarized as $\overline{J}(q) \in \mathbb{R}^{n \times n}$ and presents the corresponding invertible Jacobian matrix.

3.3.2 Decoupled task-space Equation of Motion

With the velocities v already being in the decoupled task-space, one can further lead the EoM into the decoupled task-space. The transformation is done by mapping the Jacobian matrix $\overline{J}(q)$, according to [22].

$$v = \overline{J}(q)\dot{q} \tag{3.19}$$

$$\dot{v} = \overline{J}(q)\ddot{q} + \dot{\overline{J}}(q)\dot{q} \tag{3.20}$$

assuming that $\overline{J}(q)$ is invertible, the relation

$$\ddot{q} = \overline{J}^{-1}(q) \cdot \left(\dot{v} - \dot{\overline{J}}(q)\dot{q}\right) \tag{3.21}$$

follows, which is then inserted into (3.1). The result is the EoM described in the decoupled task-space

$$\Lambda(q)\dot{v} + \mu(q,\dot{q})v + \rho = \overline{J}(q)^{-T}(\tau + \tau_{ext}), \tag{3.22}$$

where $\Lambda(q) \in \mathbb{R}^{m_i \times m_i}$ specifies the new inertia matrix in decoupled task-space. Due to the transformation, the structure of $\Lambda(q)$ equals to

$$\Lambda(q) = \overline{J}(q)^{-T} M(q) \overline{J}(q)^{-1} = diag(\Lambda_1(q), \cdots, \Lambda_r(q)), \tag{3.23}$$

which has a block-diagonal structure with symmetric, positive definite inertia matrices for each subtask $i = 1...r$. $\mu(q, \dot{q})$ describes the Coriolis/centrifugal terms in the decoupled task-space and is composed of

$$\mu(q, \dot{q}) = \overline{J}(q)^{-T}\Big(C(q, \dot{q}) - M(q)\overline{J}(q)^{-1}\dot{\overline{J}}(q, \dot{q})\Big)\overline{J}(q)^{-1} \tag{3.24}$$

and ρ defines the gravitational forces in decoupled task-space, composed of

$$\rho = \overline{J}(q)^{-T} g(q) \tag{3.25}$$

While (3.23) is decoupled, (3.24) still exhibits dynamic coupling terms between hierarchy levels.

3.3.3 Control Law

The general control law can be defined as

$$\tau = g + \tau_\mu + \sum_{i=1}^{r} N_i J_i^T F_{i,ctrl}. \tag{3.26}$$

In the previous chapter it is mentioned that just the inertia matrix within the decoupled task-space is being decoupled, while the Coriolis- and centrifugal forces still remain coupled. Therefore, τ_μ in the control law serves to decouple also these terms by compensating the outer-diagonal terms with

$$\tau_\mu = \sum_{i=1}^{r}\left(\overline{J_i}^T\left(\sum_{j=1}^{i-1}\mu_{i,j}v_j + \sum_{j=i+1}^{r}\mu_{i,j}v_j\right)\right) \tag{3.27}$$

The formulation (3.26) of the control law also includes the compensation of the gravitational terms, as well as the implementation of the control forces for each hierarchical level.

Considering the transformations and the insertion of these into (3.1), the EoM results in the decoupled task-space:

$$\Lambda_i \dot{v}_i + \mu_{i,i} v_i = F_{i,ctrl} + F_{v_i}^{ext} \tag{3.28}$$

The control forces $F_{i,ctrl}$ are adapted to each hierarchical level and are therefore also defined individually in each case $i = 1...r$, whereas $F_{v_i}^{ext}$ gives information about the external forces, acting onto the system.

3.4 Retransformation to original task-space

The reconversion is performed because the decoupled task-space velocities, as their name suggests, have the advantage of internal decoupling, but they are rather unintuitive to read and understand. Therefore this section deals with the back-conversion to the original task-space.

3.4.1 Hierarchical decoupled task-space coordinates and original task-space coordinates

Also during transformation to the original task-space, the rule of hierarchical strictness applies as well ensuring that tasks of lower priority do not influence tasks of higher priority. This is applied using a lower triangular matrix $B(q) \in \mathbb{R}^{n \times n}$ with corresponding submatrices $B_{i,j}(q) \in \mathbb{R}^{m_i \times m_i}$, describing the subtasks. Thereby the lower triangular matrix is defined as follows:

$$B(q) = \overline{J}(q) J_r^{aug}(q)^{-1}, \tag{3.29}$$

leading to the relation

$$\begin{pmatrix} v_1 \\ \vdots \\ v_r \end{pmatrix} = B(q) \begin{pmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_r \end{pmatrix}. \tag{3.30}$$

Taking a closer look at the matrix $B(q)$, the following structure can be observed

$$B(q) = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ a_{21} & I & 0 & \cdots & 0 \\ a_{31} & a_{32} & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ a_{n1} & a_{n2} & a_{n3} & \cdots & I \end{bmatrix} \tag{3.31}$$

indicating that (3.31) exactly provides the attribute for consistent decoupling.

By defining the relationship between decoupled and original task-space velocities as a function, the level-specific velocities can be expressed as a function of the original task-space velocities, demonstrated as follows:

$$v_i = \dot{x}_i + \sum_{j=1}^{i-1} B_{i,j}(q) \dot{x}_j. \tag{3.32}$$

One can see that based on the structure (3.31) of $B(q)$, the main task is only multiplied by the identity matrix and therefore can be extracted from the summation.

The relation between the accelerations is obtained using the same procedure, whereby (3.32) is derived and results in

$$\dot{v}_i = \ddot{x}_i + \sum_{j=1}^{i-1} \Big(B_{i,j}(q) \ddot{x}_j + \dot{B}_{i,j}(q, \dot{q}) \dot{x}_j \Big). \tag{3.33}$$

3.4.2 Hierarchical decoupled task-space forces and original task-space forces

The external forces acting in the decoupled task-space are denoted by $()_{v_i}$, while forces arising in the original task-space are indicated by $()_{\dot{x}_i}$. They are related by the mapping

matrix $E(q) \in \mathbb{R}^{n \times n}$ for all hierarchical levels $i = 1...r$,

$$\begin{pmatrix} F_{v_1}^{ext} \\ \vdots \\ F_{v_r}^{ext} \end{pmatrix} = E(q) \begin{pmatrix} F_{\dot{x}_1}^{ext} \\ \vdots \\ F_{\dot{x}_r}^{ext} \end{pmatrix} \tag{3.34}$$

which is the inverse of the $B(q)$-matrix (3.31),

$$E(q) = B(q)^{-T} \tag{3.35}$$

i.e. $E(q)$ is an upper triangular matrix and the force distribution over the hierarchical levels is reversed. The main task is affected by the external force and thus also by the cross coupling to the lower priority tasks. Though, e.g., task 2 is not disturbed by task 1.

Similarly as in (3.32), the relation between the decoupled task-space external forces can be represented as a function of the original task-space external forces for all hierarchy levels $i = 1...r$

$$F_{v_i}^{ext} = F_{\dot{x}_i}^{ext} + \sum_{j=i+1}^{r} E_{i,j}(q) F_{\dot{x}_j}^{ext}. \tag{3.36}$$

The relation to the original task-spaced external torques is established via

$$\tau^{ext} = J_r^{aug}(q)^T \begin{pmatrix} F_{\dot{x}_1}^{ext} \\ \vdots \\ F_{\dot{x}_r}^{ext} \end{pmatrix}. \tag{3.37}$$

3.4.3 Hierarchical decoupled task-space control law and original task-space control law

Using the formulas from the previous Sections 3.4.1 and 3.4.2, the EoM can be expressed in the original task-space by substituting in (3.1). For the hierarchical levels $i = 1...r$, this results in

$$\begin{aligned} \Lambda_i \dot{v}_i + \mu_{i,i} v_i = \Lambda_i \left(\ddot{x}_i + \sum_{j=1}^{i-1} \left(B_{i,j}(q) \ddot{x}_j + \dot{B}_{i,j}(q, \dot{q}) \dot{x}_j \right) \right) + \\ \mu_{i,i} \left(\dot{x}_i + \sum_{j=1}^{i-1} B_{i,j}(q) \dot{x}_j \right) = F_{i,ctrl} + F_{v_i}^{ext} \end{aligned} \tag{3.38}$$

The controlling force varies depending on the hierarchical level. For the main task $i = 1$

$$F_{1,ctrl} = \Lambda_1 \ddot{x}_1 + \mu_{1,1} \dot{x}_1 \tag{3.39}$$

results. Accordingly, the second most important task $i = 2$ consists of the following:

$$F_{2,ctrl} = \Lambda_2 \ddot{x}_2 + \mu_{2,2} \dot{x}_2 + \underbrace{\Lambda_2 B_{2,1}(q) \ddot{x}_1^{aug}}_{\Psi_{2,1}} + \underbrace{\Lambda_2 \dot{B}_{2,1}(q, \dot{q}) \dot{x}_1^{aug} + \mu_{2,2} B_{2,1}(q) \dot{x}_1^{aug}}_{\Gamma_{2,1}} \tag{3.40}$$

The same applies to the third most important task $i = 3$ with

$$F_{3,ctrl} = \Lambda_3\ddot{x}_3 + \mu_{3,3}\dot{x}_3 + \underbrace{\Lambda_3 B_{3,1}(q)\ddot{x}_1^{aug}}_{\Psi_{3,1}} + \underbrace{\Lambda_3 B_{3,2}(q)\ddot{x}_2^{aug}}_{\Psi_{3,2}} + \underbrace{\Lambda_3\dot{B}_{3,1}(q,\dot{q})\dot{x}_1^{aug} + \mu_{3,3}B_{3,1}(q)\dot{x}_1^{aug}}_{\Gamma_{3,1}} + \underbrace{\Lambda_3\dot{B}_{3,2}(q,\dot{q})\dot{x}_2^{aug} + \mu_{3,3}B_{3,2}(q)\dot{x}_2^{aug}}_{\Gamma_{3,2}} \tag{3.41}$$

the same way so on, until $i = r$.

All terms involving the lower triangular matrix $B(q)$ and its derivative $\dot{B}(q)$ may be grouped into $\Gamma_{i,j}(q,\dot{q}) \in \mathbb{R}^{m_i \times m_i}$ and $\Psi_{i,j}(q) \in \mathbb{R}^{m_i \times m_i}$ respectively, as already seen in the formulas (3.40) and (3.41). Since the main task must not be disturbed by tasks with lower priorities, the definition of the control force for the main task does not include any cross-couplings, respectively.

A new term $\gamma_i(q,\dot{q}) \in \mathbb{R}^{m_i \times 2\sum_{j=1}^{i-1} m_j}$ is defined, which includes $\Gamma_{i,j}(q,\dot{q})$ and $\Psi_{i,j}(q)$ and thus summarizes the EoM in the original task-space into

$$\Lambda_i\ddot{x}_i + \mu_{i,i}\dot{x}_i + \gamma_i(q,\dot{q})\begin{pmatrix} \dot{x}_{i-1}^{aug} \\ \ddot{x}_{i-1}^{aug} \end{pmatrix} = F_{i,ctrl} + F_{v_i}^{ext} \tag{3.42}$$

with

$$\gamma_i(q,\dot{q}) = (\Gamma_{i,1}, ..., \Gamma_{i,i-1}, \Psi_{i,1}, ..., \Psi_{i,i-1}), \tag{3.43}$$

consisting of

$$\Gamma_{i,j}(q,\dot{q}) = \mu_{i,i}B_{i,j} + \Lambda_i\dot{B_{i,j}} \tag{3.44}$$

and

$$\Psi_{i,j}(q) = \Lambda_i\dot{B_{i,j}}. \tag{3.45}$$

The structure of (3.43) reflects the cross-couplings of top-down disturbances between the hierarchical levels $i = 1...r$.

4 Addressing Endeffector Force Maximization through Optimization

This chapter introduces the subject and procedure of the contact-based application between the aerial system and the unknown environment. The aim is to maximize the force exerted by the endeffector when a contact situation arises.

The examined aerial system is redundant, equipped with n DoF. The existence of additional DoF enables a variety of movements and positions to perform a specific task. Due to the versatility of possible solutions for the given problem, it is not clear what the best solution for the given task might be, which makes the application of optimization problems meaningful. Referring to [23], optimization is commonly described as the search for the best possible point (optimal solution) within a decision space, aiming to achieve a specific objective.

In this context, given a redundant robot system and various constraints, the use of an optimizer offers advantages. It enables the efficient identification of the best joint positions or motion sequences to achieve the maximum exerted force of the endeffector, taking into account the given constraints. Both equality and inequality constraints can be considered.

In summary, the objective of this chapter is to design the optimization problem and necessary constraints that serve as the basis for maximizing the performance of the endeffector in a redundant robot system. Throughout this work, only the static case of the system is considered and the optimizer is exclusively used offline. Despite the current focus on the static case, the relevant constraints and conditions are carefully analyzed to provide a solid foundation for optimizing the endeffector force and to provide a starting point for future dynamic analyses of the system.

4.1 Static optimization problem

In accordance with [23], the standardized form for an optimization problem is as follows:

$$\min_{x \in \mathcal{X}} \quad f(x) \qquad \text{cost function} \tag{4.1}$$

$$\text{subject to} \quad g_i(x) = 0, \text{ for i} = 1...\text{p} \qquad \text{equality constraint} \tag{4.2}$$

$$h_i(x) \leq 0, \text{ for i} = 1...\text{p} \qquad \text{inequality constraint} \tag{4.3}$$

In this context, $g_i(x)$ describes an equation constraint and $h_i(x)$ specifies an inequality constraint. Depending on the restrictions imposed, the problem is called unconstrained or constrained, respectively. $\mathcal{X}$ represents the admissible set with which the problem can be solved, whereby $\mathcal{X}$ must not being the empty set, otherwise no solution for the problem exists.

The cost function (4.1), which here is defined as a minimization task, can be transferred to a maximization task using

$$\max_{x \in \mathcal{X}} \quad f(x) = -\Big(\min_{x \in \mathcal{X}} - f(x)\Big). \tag{4.4}$$

Based on the specification of the cost function in (4.1), the equality constraint (4.2) or inequality constraint (4.3), the optimization problem is classified into different categories e.g. linear, quadratic, non-linear and many more.

4.2 Adapted Static Optimization for Enhanced Endeffector Force

Considering the specific application, adapted static optimization refers to developing a customized solution to maximize the exerted force of the endeffector i.e. the cost function (4.5) is therefore given by the maximization of the endeffector force.

The nature of the robotic system itself creates constraints based on the mechanical properties of the system. These properties define minimum and maximum limits for the generalized coordinates q_i (4.7) and the applied torques τ_i (4.8), for $i = 1...n$. These constraints are not only the result of the physical structure of the robot, but also the systematic requirements for the safety, performance and lifetime of the system. In accordance with the goal of maximizing the force at the endeffector, the upper and lower limits for the force are defined deliberately. The lower limit is defined as zero to ensure that negative values are excluded and the upper limit i.e. the maximum possible force is not defined, which corresponds to the unbounded condition, see (4.9).

One important aspect to be considered when solving the cost function is reaching the desired end position of the endeffector (4.11), i.e. the current position should correspond to the desired position. The dynamics of the system must also be included in the optimization. Since only the static case is considered, the EoM (4.10) is simplified accordingly. The same reason also applies to the justification why velocity and acceleration limits of the generalized coordinates are not considered. Since the measurement of the force at the endeffector should be performed based on the world coordinate system, the EoM includes the change of the reference coordinate system through the rotation matrix $R \in \mathbb{R}^{3\times3}$ mapping. By assuming that the arm of the endeffector will be stretched when the maximum possible force is exerted, a certain manipulability (4.12) must be given in order to avoid a possible singularity.

$$\begin{aligned}
\text{maximize} \quad & |Fe| \in \mathbb{R}^{6\times1} && (4.5)\\
\text{w.r.t.} \quad & q_d, \tau, Fe && (4.6)\\
\text{subject to} \quad & q_{min_i} < q_i < q_{max_i} \text{ for i} = 1...\text{n} && (4.7)\\
& \tau_{min_i} < \tau_i < \tau_{max_i} \text{ for i} = 1...\text{n} && (4.8)\\
& 0 < Fe && (4.9)\\
& g(q_d) = \tau + J(q_d)^T \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} Fe && (4.10)\\
& x_d = f(q_d) && (4.11)\\
& m(q_d) \geqslant 0.1 && (4.12)
\end{aligned}$$

All of the above mentioned conditions combined result in the definition of the non-linear static optimization task, see (4.5) to (4.12). The non-linearity occurs in the equality constraints (4.10) and (4.11), since the forward kinematics of the robotic system is used to minimize the error between current and desired position, as well as the use of the EoM, which requires the nonlinear Jacobian matrix to convert between force and torque.

The inequality constraint, which describes the minimum required manipulability $m(q_d)$ of the system has been set to 0.1, based on Paper [24], which set this requirement to minimal value of 0.08.

Referring to (4.1), the elements in x within the resulting solution set $\mathcal{X}$ are consequently composed of the selection of generalized coordinates, the torque, and the resultant maximum force at the endeffector.

4.3 Hypothesized Expectations

The current assumption based on the cost function and the defined constraints indicates that the joint positions of the robot could tend to have an elongated orientation in order

to reach the desired end position. Furthermore, it is assumed that the aerial base is tilted in the direction in which the gravitational forces act to amplify the force. Thus, if a pose is defined with positive value along the x-axis position, the base should be oriented in such a way that gravity is used to act in this direction. An illustrative result of this alignment is shown in Figure 4.1.

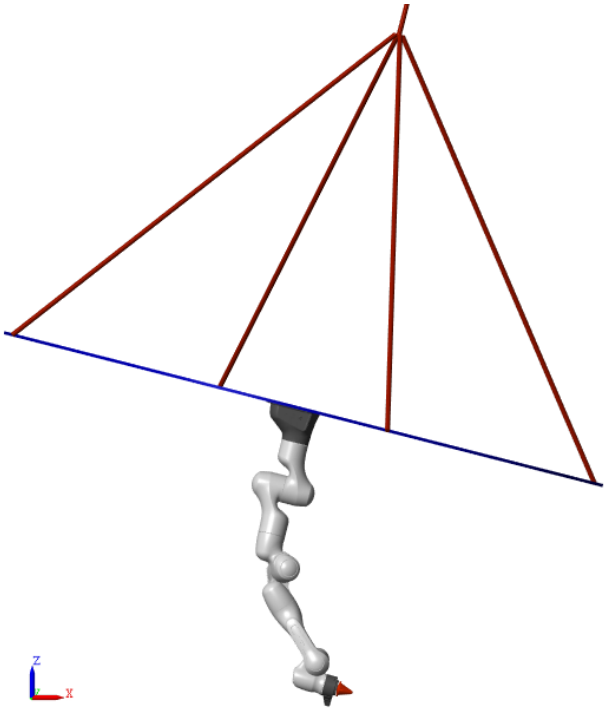


Figure 4.1: Hypothesized Expectation of the pose of the Suspended Aerial Manipulator

This assumption is based on the hypothesis that a more elongated positioning would allow maximizing the force exerted at the endeffector by taking advantage of gravity. The picture of an expected joint position is currently based only on this hypothesis and requires further investigation to be confirmed or refused. It is a theoretical prediction that needs to be verified by future analysis and testing before conclusions can be drawn, which will be discussed in Chapter 5.

5 Experimental Approach

Within this chapter, the simulative implementation is performed based on the theories and concepts elaborated in the previous parts. The tools and platforms used include MATLAB version 2022b and Maple version 2022 for the mathematical calculations. The environment Simulink is used for simulations to ensure a comprehensive understanding and realistic representation.

The code structure across all mentioned programming environments provides a configuration file. This flexible structure ensures easy modification and adaptation of the robotic system itself, the parameters and the initial conditions, which facilitates the experimental investigation of different scenarios and parameter combinations.

Without any general limitation or restriction, the robot system **Franka Emika Panda** and a multirotor as aerial base are used for this master thesis, see Figure 5.1.

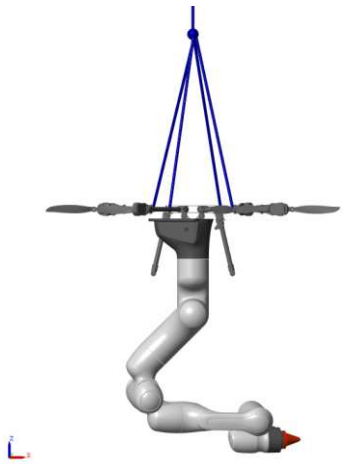


Figure 5.1: Franka Emika Panda attached on a Multirotor in its initial position

Section 5.1 is dedicated to the general information of the overall system itself, while Section 5.2 and 5.3 describe the implementation and testing of HC. Section 5.4 then deals with the implementation and testing of the optimizer. Finally, in Chapter 5.5, the controller and the optimizer are combined and comprehensively tested in an use-case defined within.

5.1 Robotic system

The SAM is composed of an aerial platform (multirotor vehicle) in combination with a robot arm (which in this case is the robotic system Franka Emika Panda) and serves as the basis for the simulations. It has a total of 10 DoF, composed of 7 DoF provided by the robot arm and 3 DoF required for positioning the platform. The whole system is attached to a crane to facilitate precise positioning.

The schematic structure of the SAM in its zero configuration is illustrated in Figure 5.2.

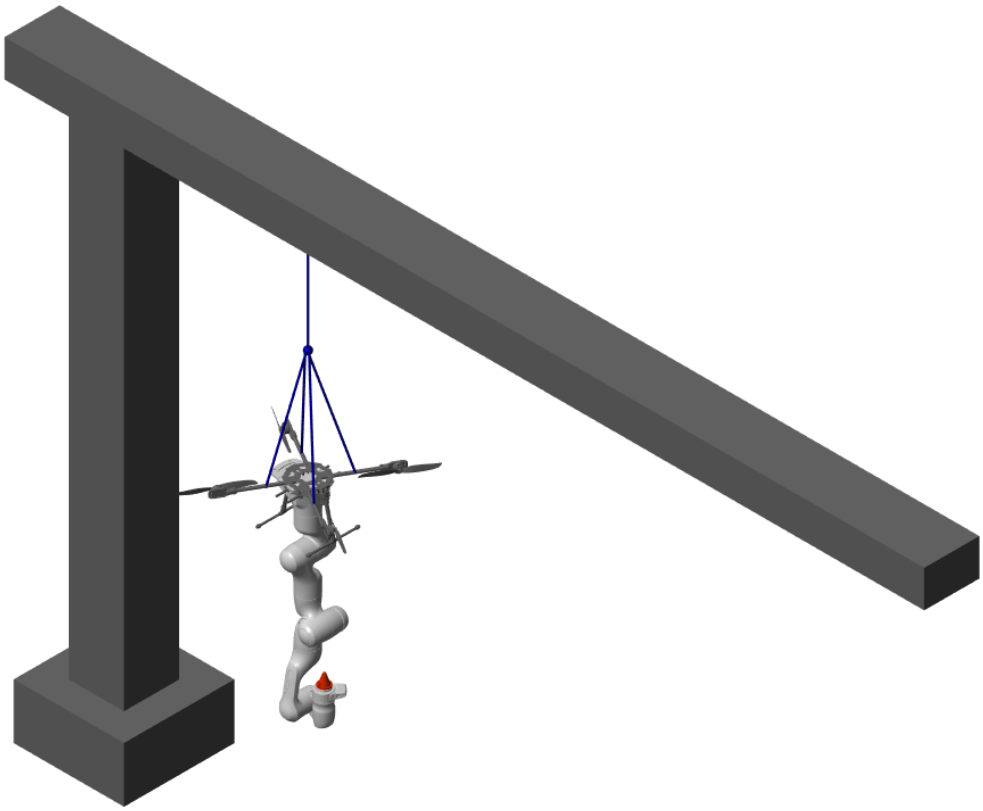


Figure 5.2: Suspended Aerial Manipulator mounted on crane

While the system is interacting with the environment, a total of 3 tasks is supposed to be carried out simultaneously. Thereby, these tasks are assigned with different priority.

The main task, which has the highest priority value starting at $i = 1$, is following a given trajectory in order to reach the desired position x_d of the endeffector. Thereby, the trajectory is described using a polynomial function of 5^{th} order. The output of this task is the actual position of the endeffector x_e in task-space, described by six coordinates in

translational and rotational manner

$$x_e = \begin{bmatrix} x \\ y \\ z \\ \varphi_e \\ \theta_e \\ \psi_e \end{bmatrix} \in \mathbb{R}^{6\times 1}. \tag{5.1}$$

The task with second priority involves the avoidance of oscillations of the aerial base and their rapid elimination, prioritized with $i = 2$. The task, denoted by x_b, is described by three coordinates defined in the task-space, which correspond to the axes of rotation of the base

$$x_b = \begin{bmatrix} \varphi_b \\ \theta_b \\ \psi_b \end{bmatrix} \in \mathbb{R}^{3\times 1}. \tag{5.2}$$

The third task consists of controlling the movement of the robot arm elbow in such a way that potential singularities, that could arise due to ambiguous movements of the arm, are avoided. With a defined priority of $i = 3$, it is the least important task and is described with one rotational coordinate x_{elbow} in the task-space

$$x_{elbow} = \begin{bmatrix} x_{elbow} \end{bmatrix} \in \mathbb{R}^{1\times 1}. \tag{5.3}$$

The applied controller is the hierarchical impedance-based whole-body controller described in detail in Section 3 and fulfills a central role. It ensures that tasks with different priorities do not interfere with each other. This important functionality enables the task with highest priority to be executed undisturbed and without interference of other tasks.

5.2 Implementation of the controller

The structure of the controller corresponds to the specifications in Chapter 3 and is shown in Figure 5.3.

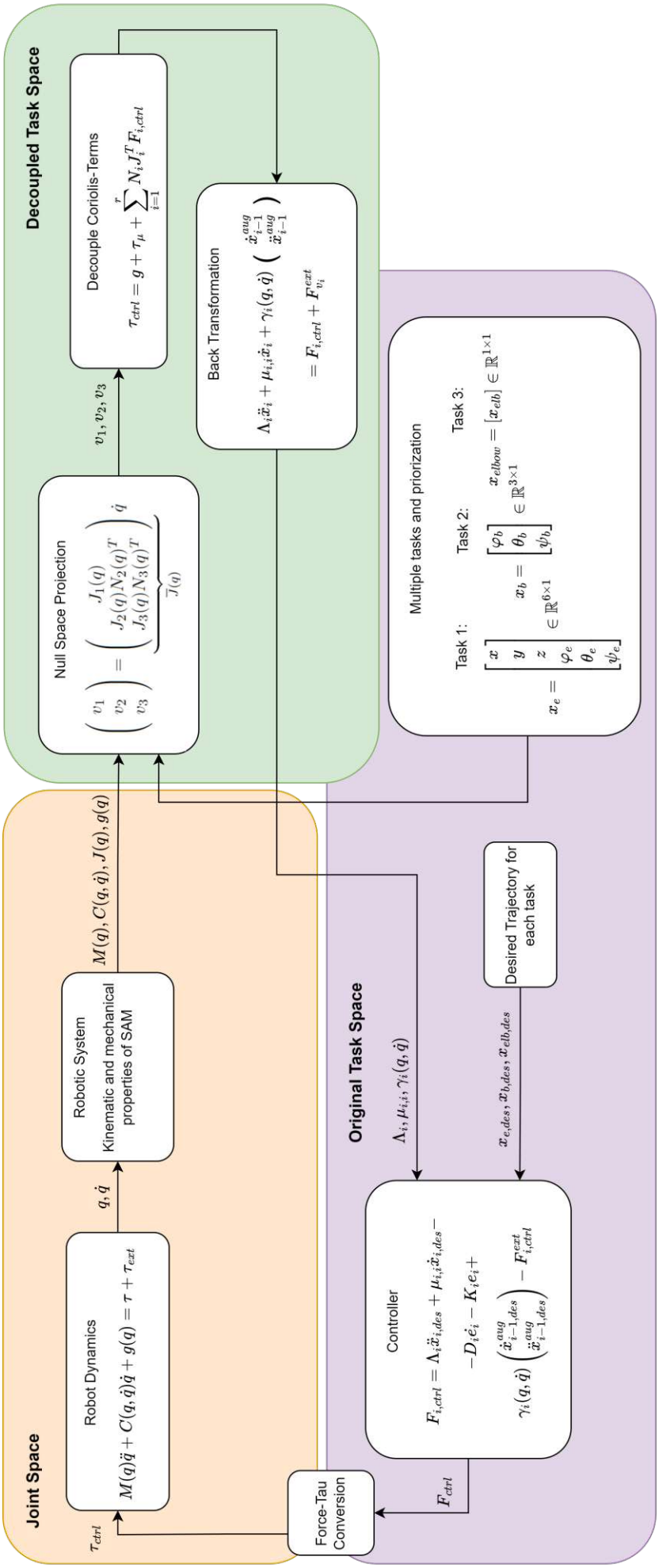


Figure 5.3: Structure of the Hierarchical Controller

The build up control task hierarchy defined in task-space comprises the three necessary tasks to be fulfilled as described in Chapter 5.1, equations (5.1), (5.2) and (5.3).

5.2.1 Incorporating Top-Down Disturbances in Equation of Motion

Based on Figure 3.1, a decoupled task-space is desired in order to prevent inter-dependencies between the different motion tasks. This transition allows for the clear definition of a strict hierarchy between the different tasks, organized according to their designated priorities. Its realization is achieved by a dynamically consistent null-space projector. This means that the joint coordinates are first transformed into the decoupled task-space in order to apply the control law there. Afterwards, an inverse-transformation into the original task-space is carried out. The augmented projection is selected as the null-space projection. The rationale behind this choice is explained in Chapter 3.2.2.

By definition in (3.16), the projector involves the augmented Jacobian matrix, which, according to (3.10), stacks all level-specific Jacobian matrices according to their priority down to $i = 3$. This leads to

$$J_1^{aug}(q) = \begin{pmatrix} J_1(q) \end{pmatrix} \tag{5.4}$$

$$J_2^{aug}(q) = \begin{pmatrix} J_1(q) \\ J_2(q) \end{pmatrix} \tag{5.5}$$

$$J_3^{aug}(q) = \begin{pmatrix} J_1(q) \\ J_2(q) \\ J_3(q) \end{pmatrix} \tag{5.6}$$

and results in the decoupled task-space coordinates for all three tasks, as in (3.18)

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \underbrace{\begin{pmatrix} J_1(q) \\ J_2(q)N_2(q)^T \\ J_3(q)N_3(q)^T \end{pmatrix}}_{\overline{J}(q)} \dot{q} \tag{5.7}$$

and consequently, obtains a hierarchically decoupled EoM, see Section 3.3.2. Following the Chapters 3.3.3 and 3.4, which include decoupling the Coriolis terms and the back-transformation to the original task-space, results in the EoM (3.42) in the original task-space, but now including the top-down disturbances summarized in $\gamma_i(q, \dot{q})$. Utilizing this formula as a foundation, the subsequent action involves the practical implementation of the controller.

5.2.2 Control force

Given [14] and equation (3.42), the applied control force $F_{i,ctrl}$ on each hierarchy level $i = 1..3$, is chosen as

$$F_{i,ctrl} = \Lambda_i \ddot{x}_{i,des} + \mu_{i,i}\dot{x}_{i,des} - D_i\dot{e}_i - K_i e_i + \gamma_i(q,\dot{q}) \begin{pmatrix} \dot{x}^{aug}_{i-1,des} \\ \ddot{x}^{aug}_{i-1,des} \end{pmatrix} - F^{ext}_{i,ctrl} \tag{5.8}$$

where $e_i = x_i(q) - x_{i,des} \in \mathbb{R}^{m_i}$ describes the errors in task-space position and $\dot{e}_i \in \mathbb{R}^{m_i}$ the error in task-space velocities, respectively. The positive definite and symmetric matrices $K_i \in \mathbb{R}^{m_i \times m_i}$ and $D_i \in \mathbb{R}^{m_i \times m_i}$ are the task-space stiffness and task-space damping coefficients and describe the desired compliance-based behavior (see Section 2.2.3).

The determination of the values of the diagonal matrices K_i and D_i are conducted empirically, considering physical criterion. The stiffness K_i adjusts the speed at which a system reacts to deviations from the target position. It should be set sufficiently large to react to external disturbances but should not be too high in order to avoid possible instability of the system. The damping D_i defines the system's ability to return to its initial position after a drift and should be significantly high to suppress oscillations. The empirically established values of K_i and D_i are listed in Appendix A, Section A.2. It is apparent that D_i is defined as $D_i = 2\sqrt{K_i}$.

The value $F^{ext}_{i,ctrl}$ in (5.8), further defined in (5.9), mitigates the effects of external forces. If measurements or estimates of these forces are available (case 2), they are eliminated. Otherwise, case 1 applies.

$$F^{ext}_{i,ctrl} = \begin{cases} 0, & \text{for case 1} \\ \sum_{j=j+1}^{r} E_{i,j}(q) F^{ext}_{\dot{x}_j} & \text{for case 2} \end{cases} \tag{5.9}$$

In case 1, no information about external forces is available, which leads to cross-coupling between the hierarchies of the tasks. This means that forces acting on task 1 also influence tasks with a lower priority. In contrast, case 2 enables these cross-couplings to be prevented, as the interference is known. As a result, disruptions that affect task 1 are not transferred to the other tasks.

The Section 5.3 involves the evaluation of the controller in various use-cases. In particular, scenarios in which external forces occur are also considered, whereby both possible cases, as in (5.9), are tested. The initial condition for the generalized coordinates and its derivatives corresponds to (A.1) and (A.2), unless otherwise specified.

5.3 Tests and evaluation of the Hierarchical Controller

The evaluation of the HC is performed within this chapter. This is done by recreating the use-cases as conducted in [18]. This comprises three different scenarios. In the first use-case, the robotic system is required to hold the desired position, which means that the desired end position corresponds to the robot's initial position. In the second scenario, the robot follows a trajectory, while in the third and fourth scenario an additional external force is exerted on the system while performing scenario 2. It is important to note that the optimizer is NOT included while testing.

The structure of each graphical evaluation of an experiment follows an uniform pattern. In the first figure of the experiment, the trajectories of the desired trajectory (red curves) and the current trajectory (blue curves) are presented. The second image illustrates the error between these 2 curves. The first 6 sub-images per result figure show the trajectory of the task with the highest priority and its representation in all six spatial coordinates. Sub-images 7 to 9 illustrate the joint coordinates of task 2, i.e. the movement of the base, and the last image presents the trajectory of the elbow. This information is additionally available in the legend of each picture.

Scenario 1: Hold initial value

Use-Case 1:
The endeffector is supposed to hold the initial position, i.e. $x_{i,des}$ is defined as follows for all tasks $i = 1$ to $i = 3$

$$x_{1,des} = x_{e,init} \in \mathbb{R}^{6\times 1}$$
$$x_{2,des} = x_{b,init} \in \mathbb{R}^{3\times 1}$$
$$x_{3,des} = x_{elb,init} \in \mathbb{R}^{1\times 1}$$

The Illustration 5.4 presents the outcomes of the first scenario. The charts show the comparison between the current (blue curves) and desired (red curves) values. Each subplot represents a specific parameter of each tasks over time. The second Figure 5.5 provides a more detailed representation of the deviations between the desired and actual values (the error) and shows, that the controller accurately holds the initial position of the robotic system.

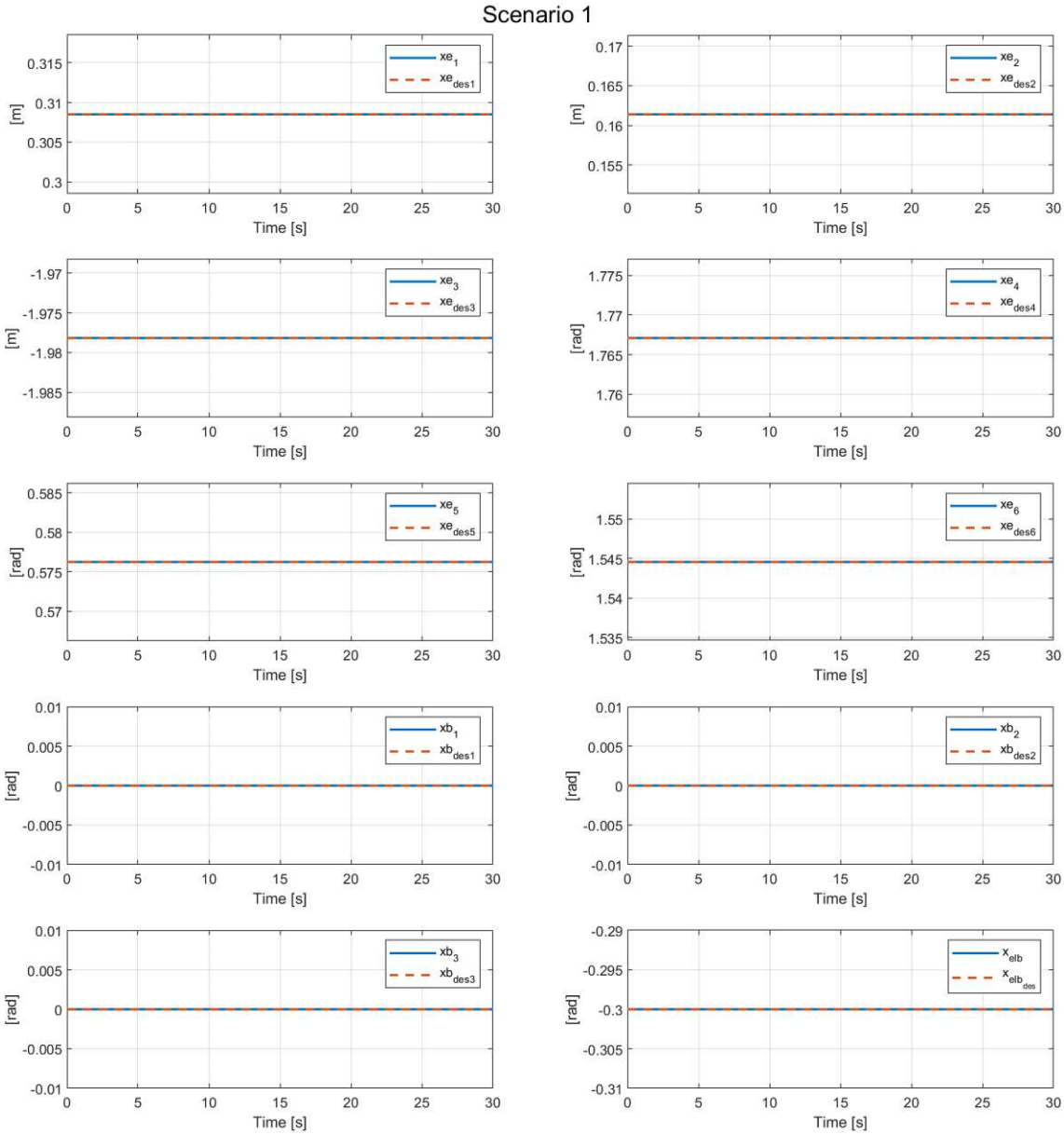


Figure 5.4: Scenario 1 Use-Case 1: Actual value compared with desired value for Task 1 to 3

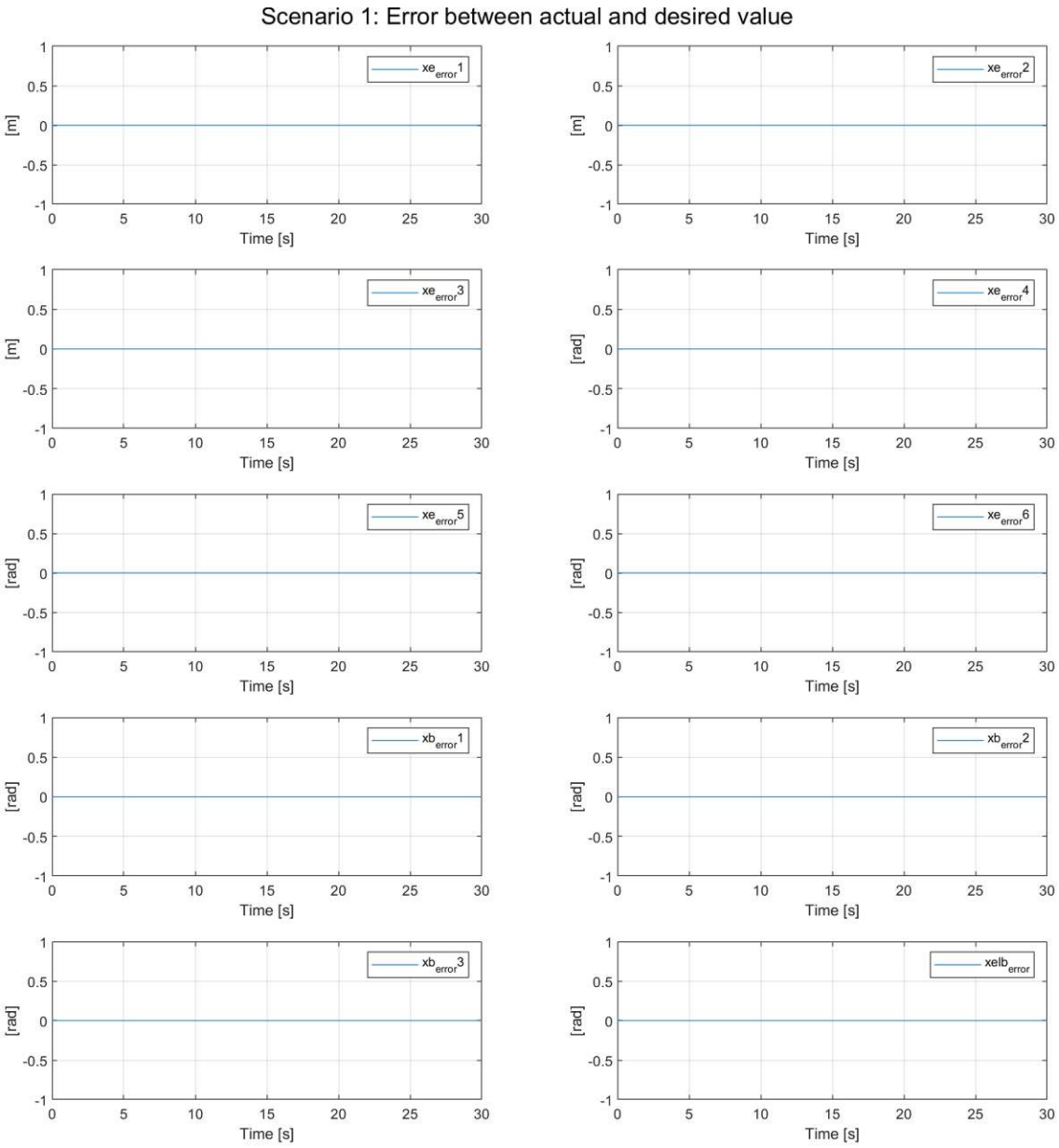


Figure 5.5: Scenario 1 Use-Case 1: Error of each parameter for Task 1 to 3

Use-Case 2:
Based on [18], the second experiment within scenario 1 is designed to maintain the initial position of the endeffector and the elbow, while the base follows the configurations below (following a polynomial function of 5^{th} order).

$$
\begin{aligned}
x_{1,des} &= x_{e,init} \in \mathbb{R}^{6\times 1} \\
x_{2,des} &= x_{b,init} + \begin{bmatrix} 0 & 0 & 10 \end{bmatrix}^T \deg \in \mathbb{R}^{3\times 1} \\
x_{3,des} &= x_{elb,init} \in \mathbb{R}^{1\times 1}
\end{aligned}
$$

The outcome of the experiment, visualized and presented in Figure 5.6, indicates the desired trajectories (red curves) as defined above, and precise following of the current values (blue curves). A more detailed illustration is provided in Figure 5.7, which shows the deviations between the desired and actual positions. It is evident that all tasks converge and that lower-priority tasks do not interfere with higher-priority tasks. i.e. the movement of the base (Task 2) does not have any effect on Task 1.

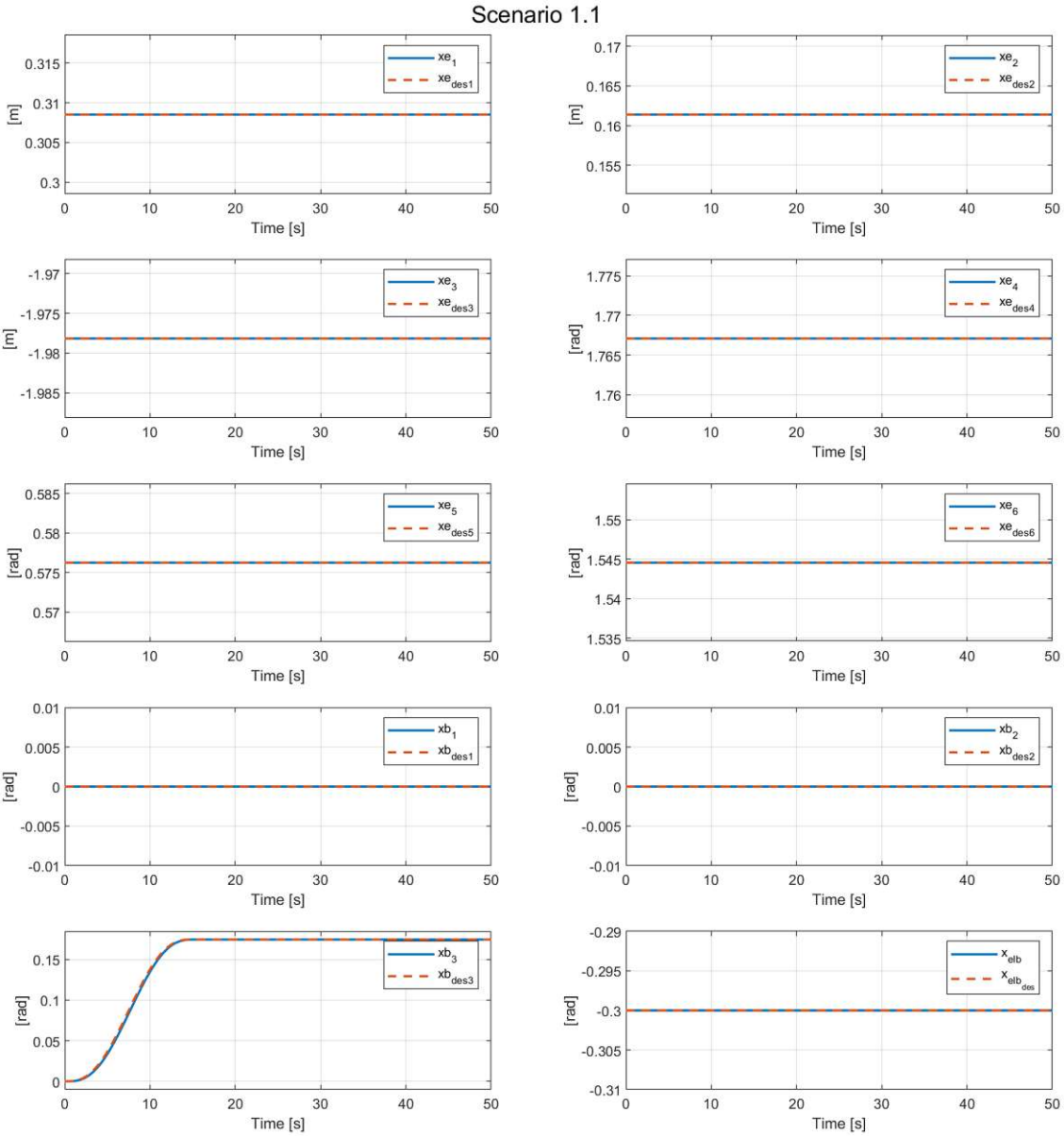


Figure 5.6: Scenario 1 Use-Case 2: Actual value compared with desired value for Task 1 to 3

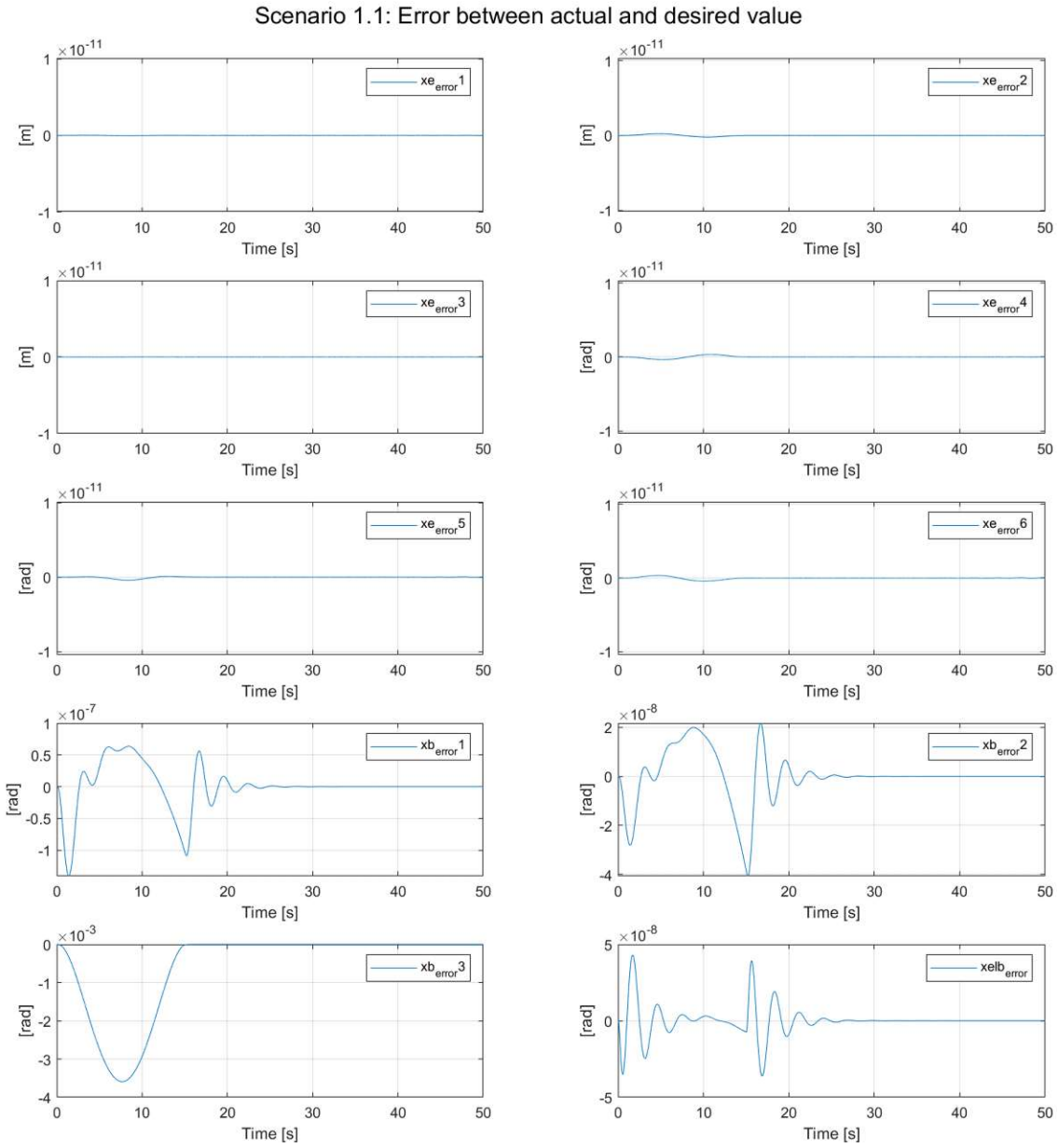


Figure 5.7: Scenario 1 Use-Case 2: Error of each parameter for Task 1 to 3

Scenario 2: Reaching given desired end-positions

The system is supposed to reach a desired position for each task $i = 1$ to $i = 3$, as defined below. The given trajectories are designed using a polynomial function of 5^{th} order.

$$x_{1,des} = x_{e,init} + \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0 & 0 & 0 \end{bmatrix}^T \in \mathbb{R}^{6\times 1}$$
$$x_{2,des} = x_{b,init} + \begin{bmatrix} 0 & 0 & 20 \end{bmatrix}^T \text{deg} \in \mathbb{R}^{3\times 1}$$
$$x_{3,des} = x_{elb,init} + 10 \text{ deg} \in \mathbb{R}^{1\times 1}$$

The simulation findings show that the controller effectively follows the desired trajectory, as can be seen in Figure 5.8 by the comparison between the desired (red curves) and current trajectories (blue curves). The tasks converge successfully towards their end position and are maintained stable. A more detailed representation can be found in Figure 5.9, which illustrates the error between the desired and actual trajectory. It is noticeable that while Task 2 (xb_{error1} to xb_{error3}) shows slight oscillations and converges more slowly, Task 1 (xe_{error1} to xe_{error6}) remains unaffected, which reflects the hierarchical behavior.

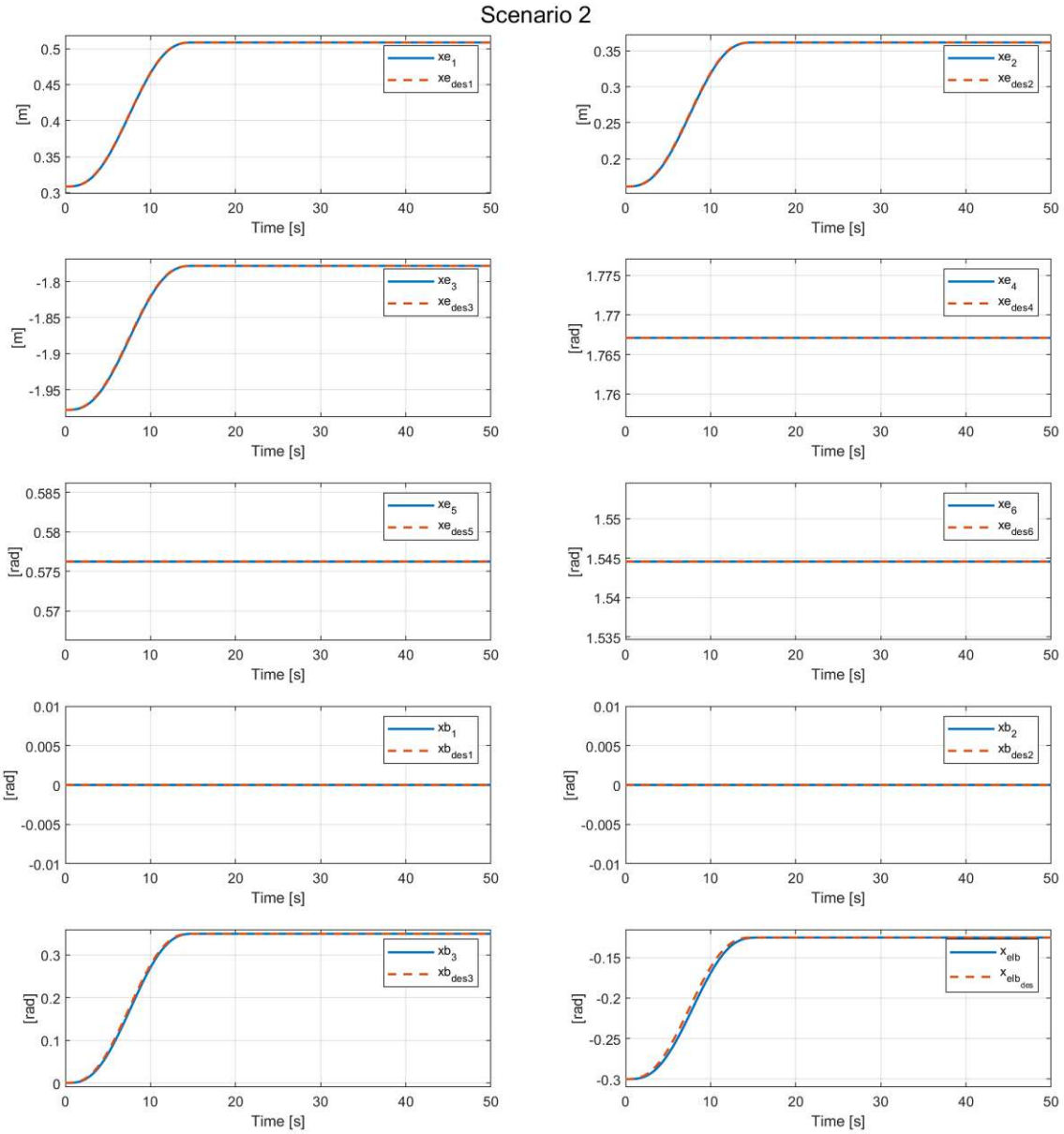


Figure 5.8: Scenario 2: Actual value compared with desired value for Task 1 to 3

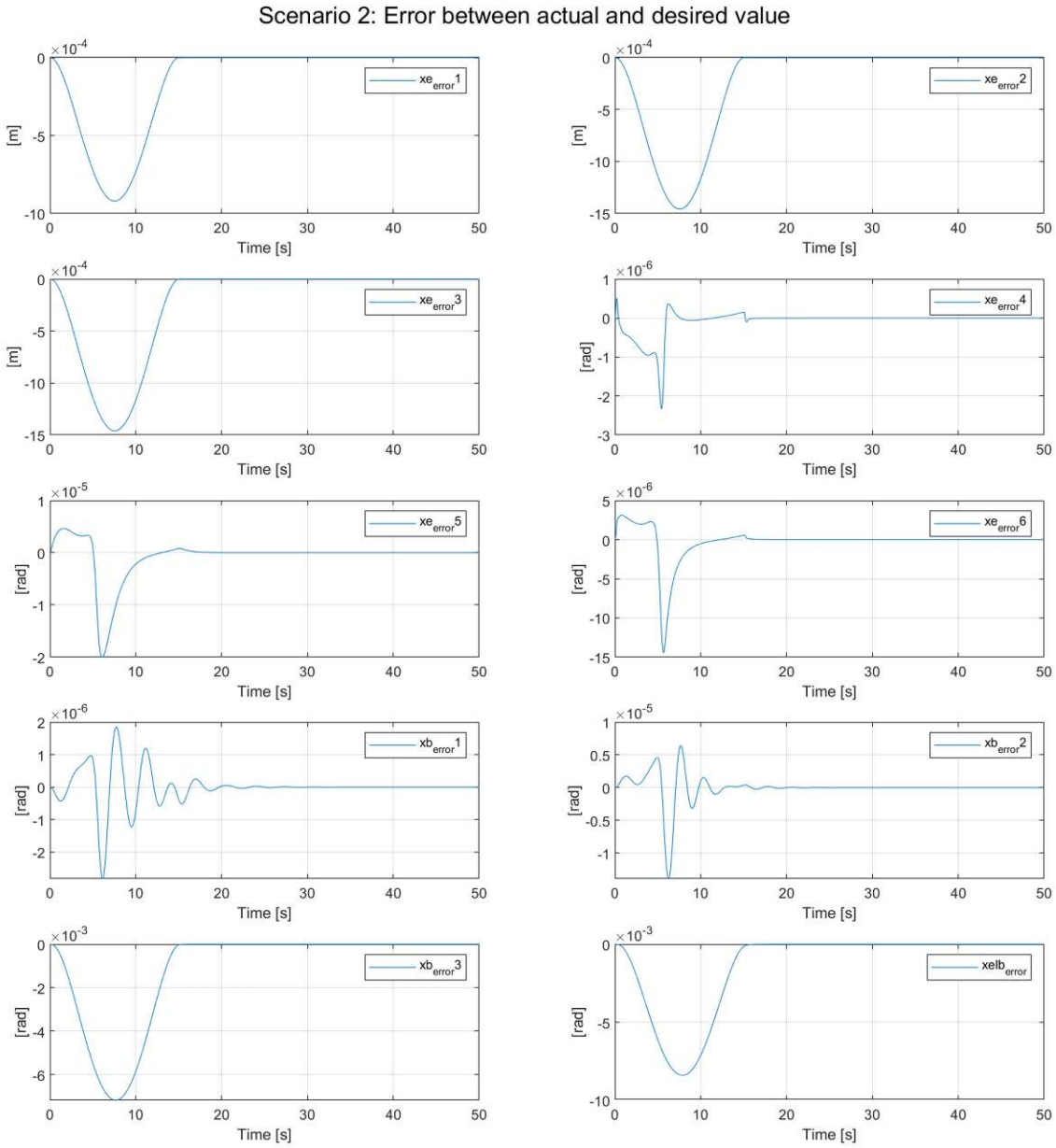


Figure 5.9: Scenario 2: Error of each parameter for Task 1 to 3

Scenario 3: Achieving Desired End-Positions in presence of External Oscillations

Scenario 3 presents a detailed analysis of the controller behavior under the influence of external forces. In this procedure, the application from Scenario 2 is repeated, whereby this time external forces occur in directions x, y and z at the endeffector, as illustrated in Figure 5.10. In addition, two different cases according to definition (5.9) are considered.

In the first instance (Use-case 1), the external force is not integrated into the feedback loop of the controller. This makes it possible to observe the system behavior if the external force is not taken into account. In contrast, the second situation (Use-case 2) actively integrates the external force into the controller's feedback loop, resulting in an adapted control strategy.

The precise examination of different scenarios provides insights into the robustness of the controller against external influences and enables a precise examination of the responsiveness of the system to different external forces in the different spatial directions.

Use-Case 1:
The solutions of Use-Case 1 illustrate that while external forces are present, the deviation between the desired trajectory (red curves) and the actual trajectory (blue curves), as shown in Figure 5.11, is particularly evident. A more precise visualization of this variance is shown in Figure 5.12, which illustrates the error between the two curves. The error during the occurrence of the external force is noticeable, nevertheless it also becomes apparent that the controller can follow the desired trajectory very quickly after the force has decayed. As a result, all tasks converge.

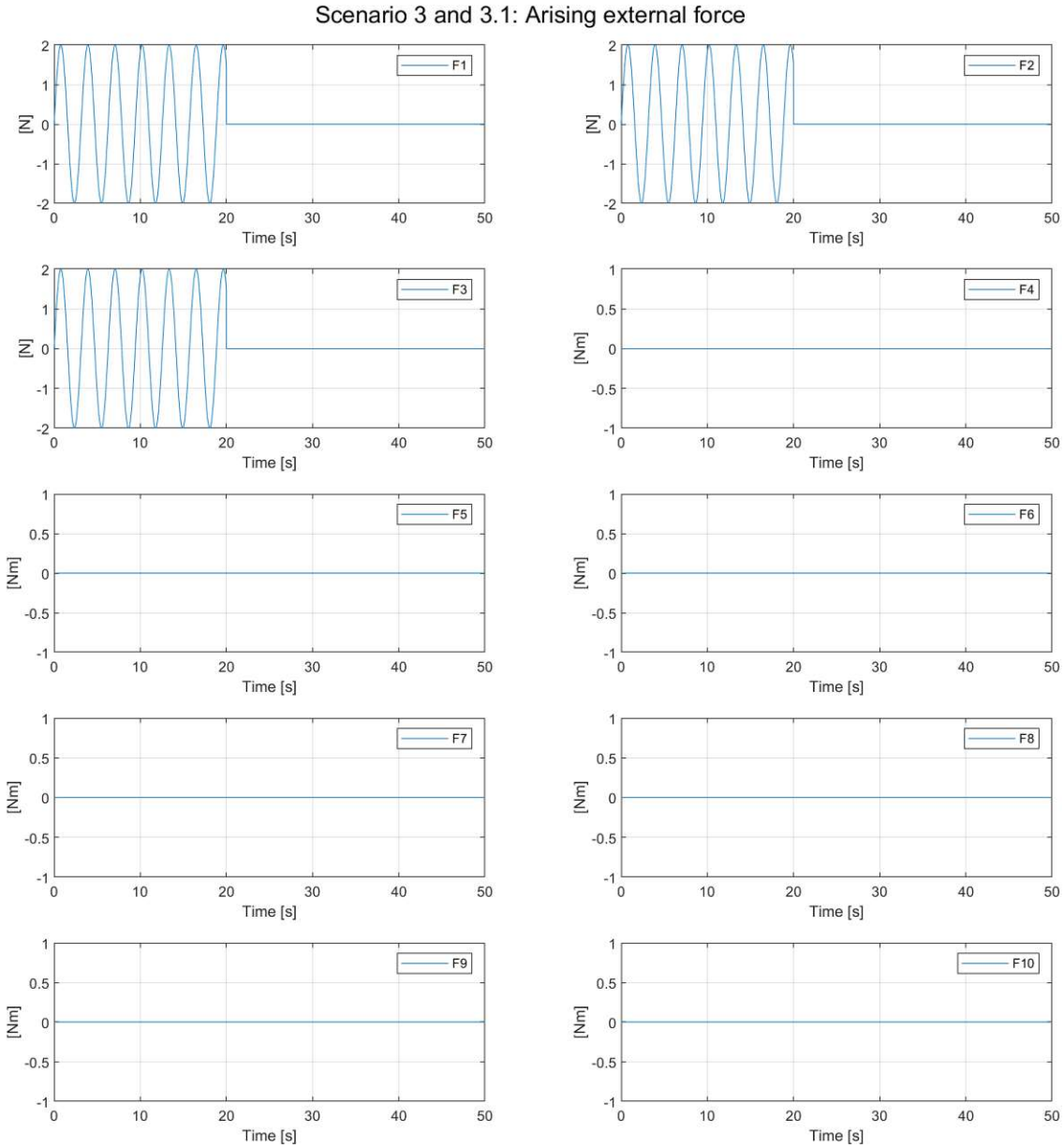


Figure 5.10: Scenario 3: External arising force

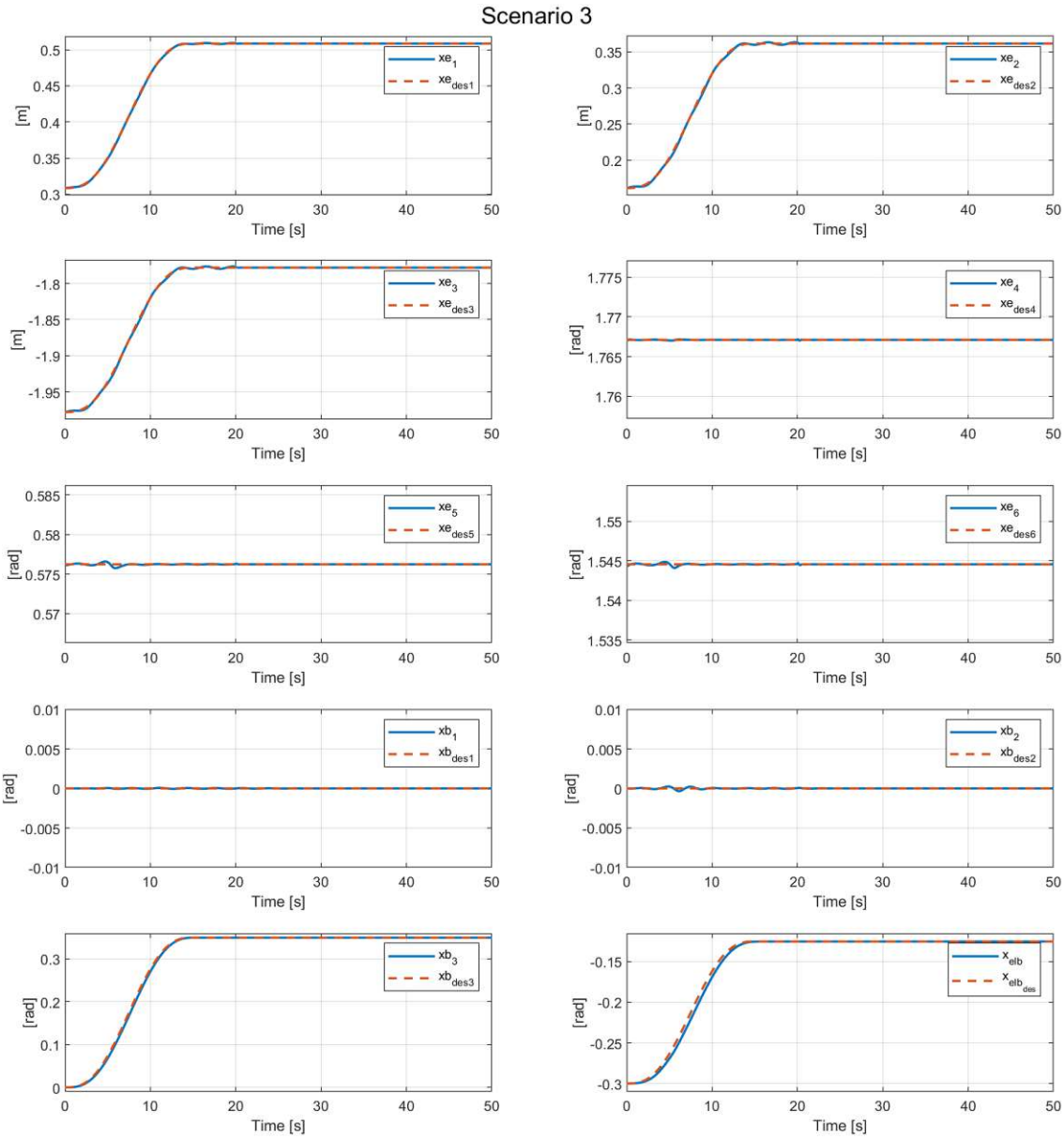


Figure 5.11: Scenario 3: Actual value compared with desired value for Task 1 to 3

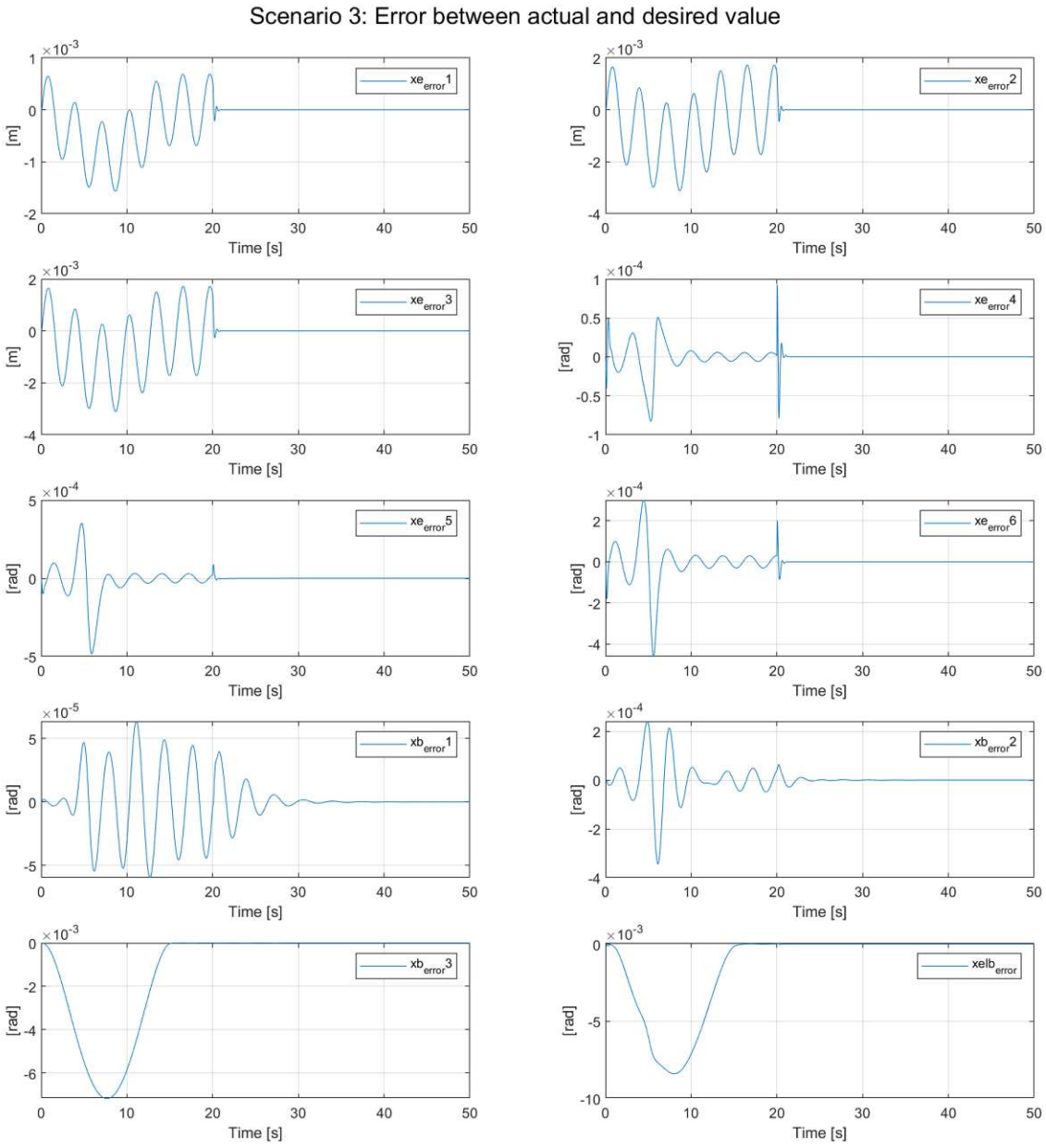


Figure 5.12: Scenario 3: Error of each parameter for Task 1 to 3

Use-Case 2:
In Use-Case 2, the feedback loop becomes active, which means that the external forces can be actively suppressed as these are now known. This is clearly illustrated in Figure 5.13, which shows a direct comparison between the desired (red curves) and the actual trajectory (blue curves). Figure 5.14, representing the error between the two curves, is particularly informative. The active suppression of the external force makes it possible to see that even when these external influences occur, there are no significant deviations.

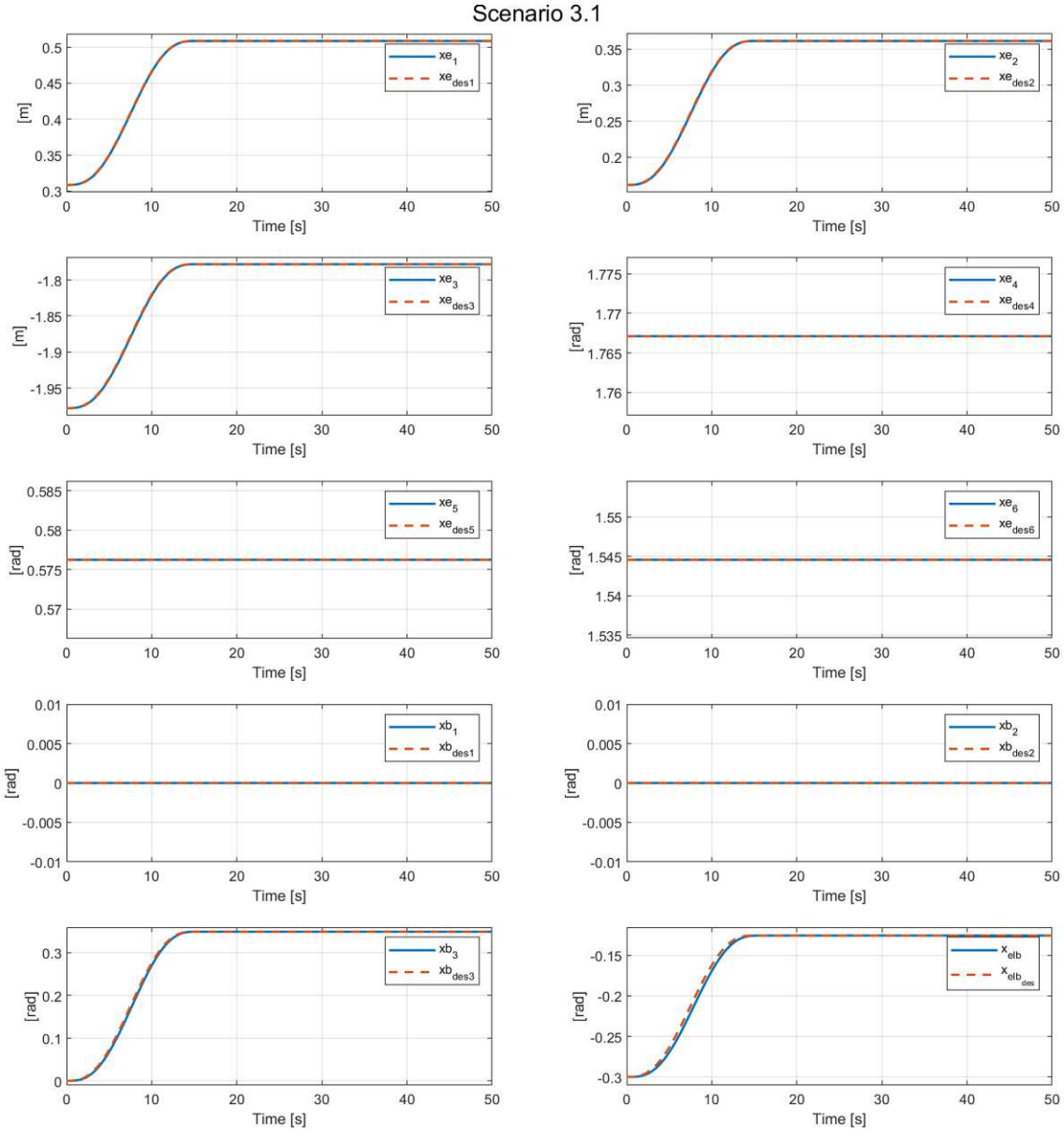


Figure 5.13: Scenario 3.1: Actual value compared with desired value for Task 1 to 3

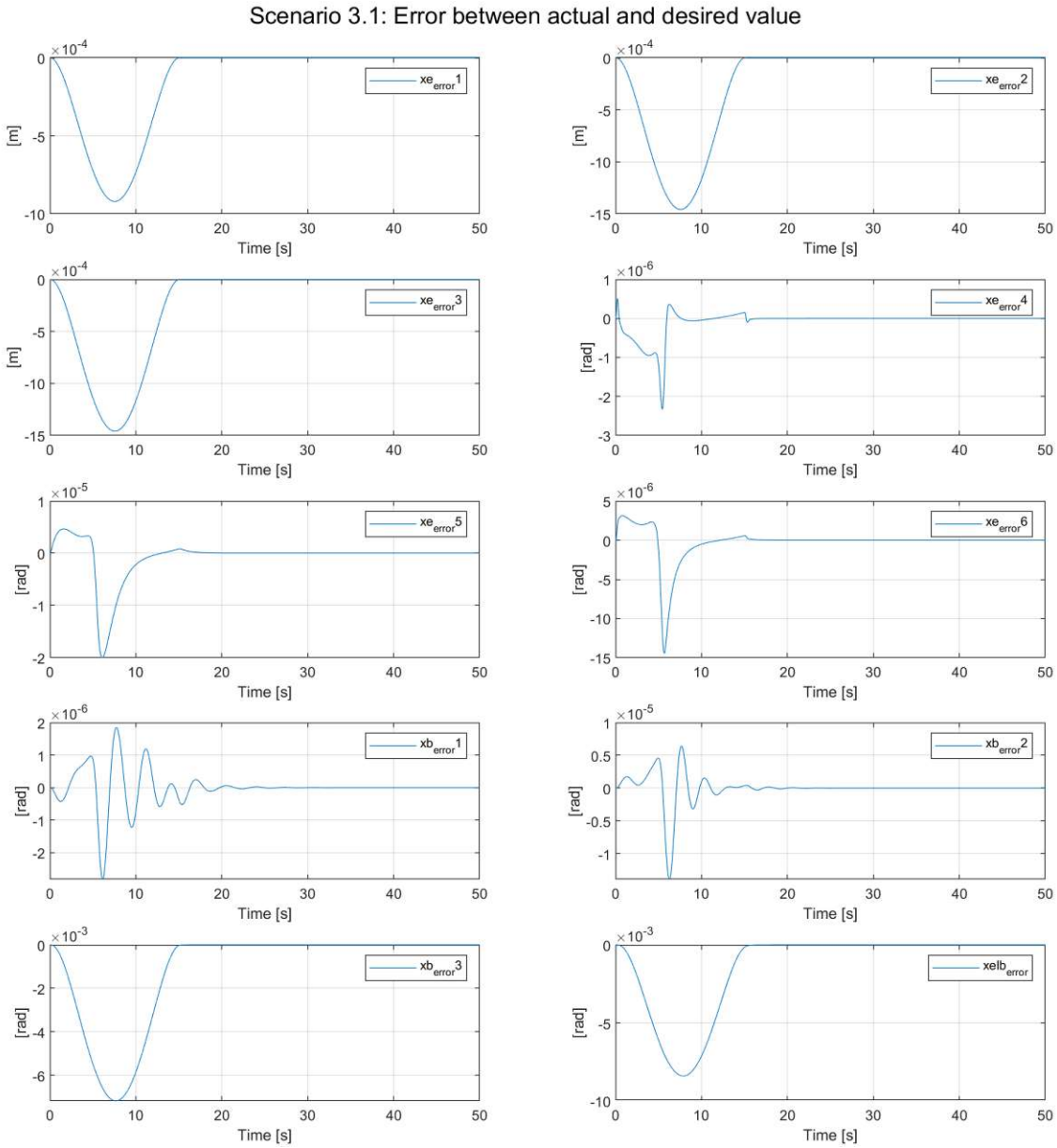


Figure 5.14: Scenario 3.1: Error of each parameter for Task 1 to 3

Scenario 4: Achieving Desired End-Positions in presence of External Forces

Scenario 4 refers to the previous studies and is characterized by a different approach. The external force is presented in a new variant, as illustrated in Figure 5.15. In contrast to the previous scenario, forces and moments now manifest themselves in all spatial directions and at different points in time. The behavior of the system and the adjustment of the HC are analyzed again, with (Case 1) and without (Case 2) a feedback loop.

Use-Case 1:
In the results of Case 1, in which no feedback loop is used, Figure 5.16 shows a comprehensive comparison between the desired (red curves) and current (blue curves) trajectories, while Figure 5.17 illustrates the respective deviations between these two curves. The behavior of the controller shows that cross-couplings occur without the use of the feedback loop. Due to the lack of knowledge about the exact magnitude of the occurring forces and torques, certain deviations result during the appearance of external influences. However, the HC regulates these deviations extremely quickly as soon as the external force decreases.

Even with a considerable external force of $200N$ in x-direction of the endeffector, the deviation remains comparatively small. This favorable characteristic enables the controller to converge all tasks effectively and efficiently, supported by the fast and stable control behavior.

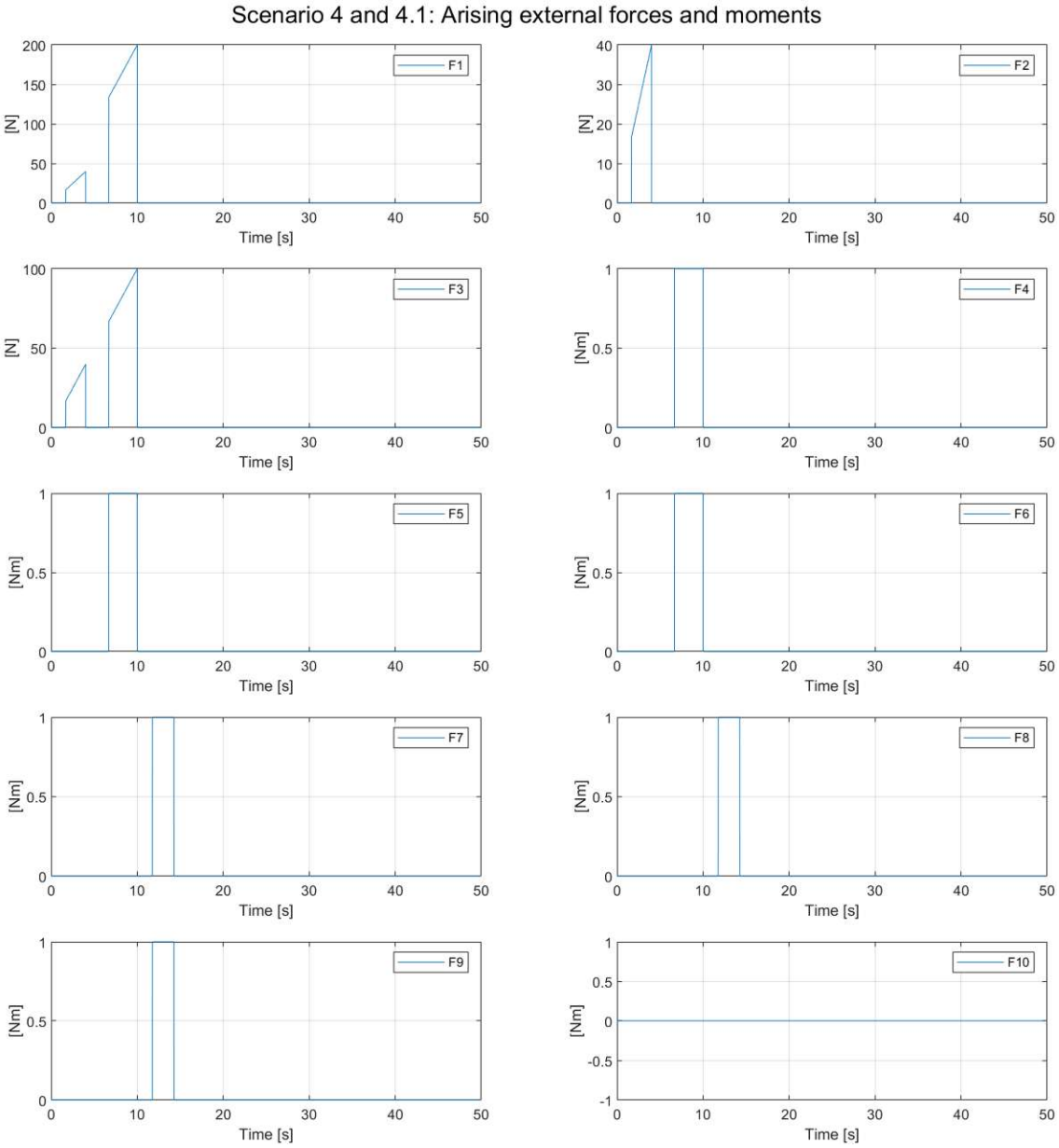


Figure 5.15: Scenario 4: External arising force

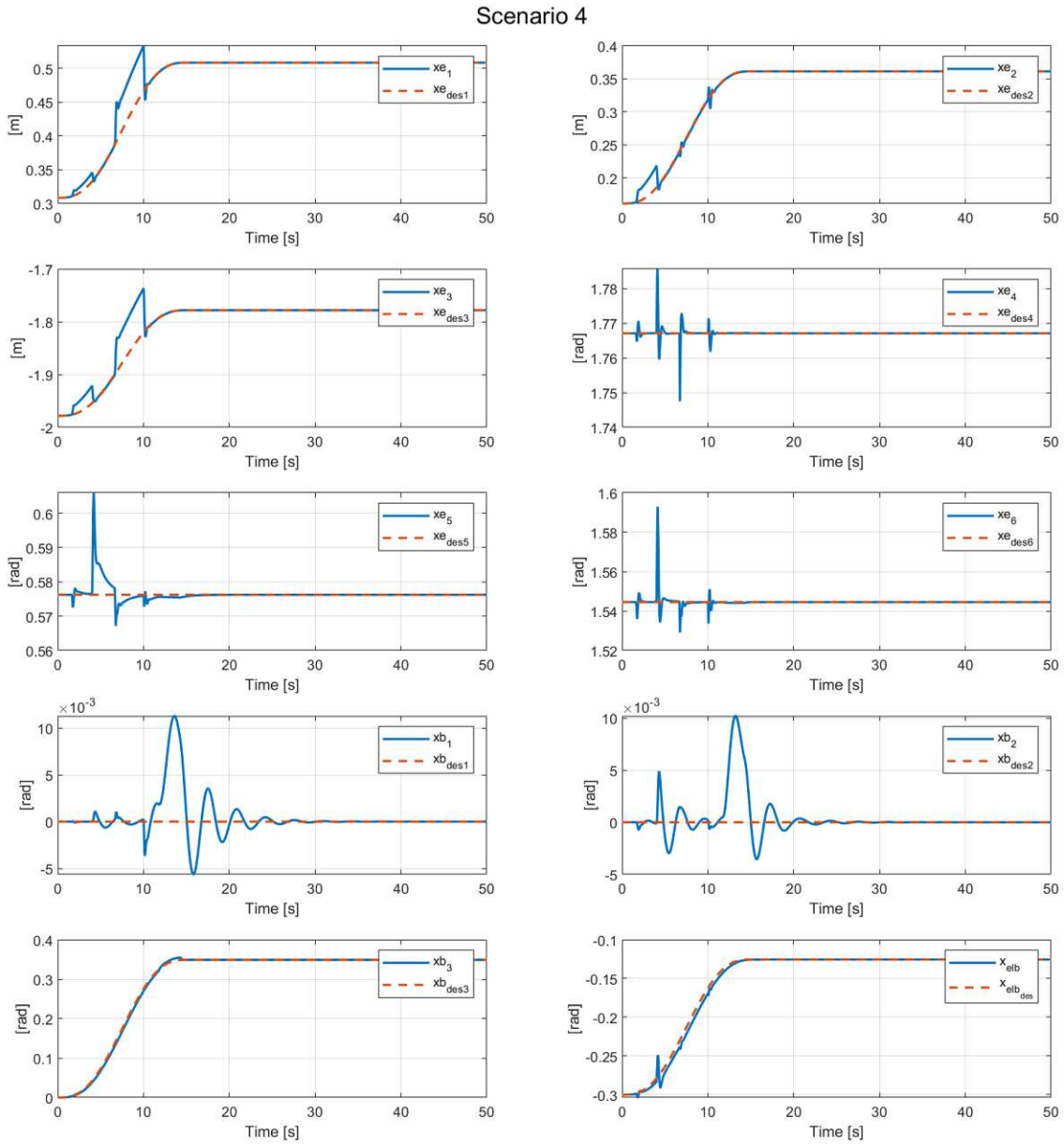


Figure 5.16: Scenario 4: Actual value compared with desired value for Task 1 to 3

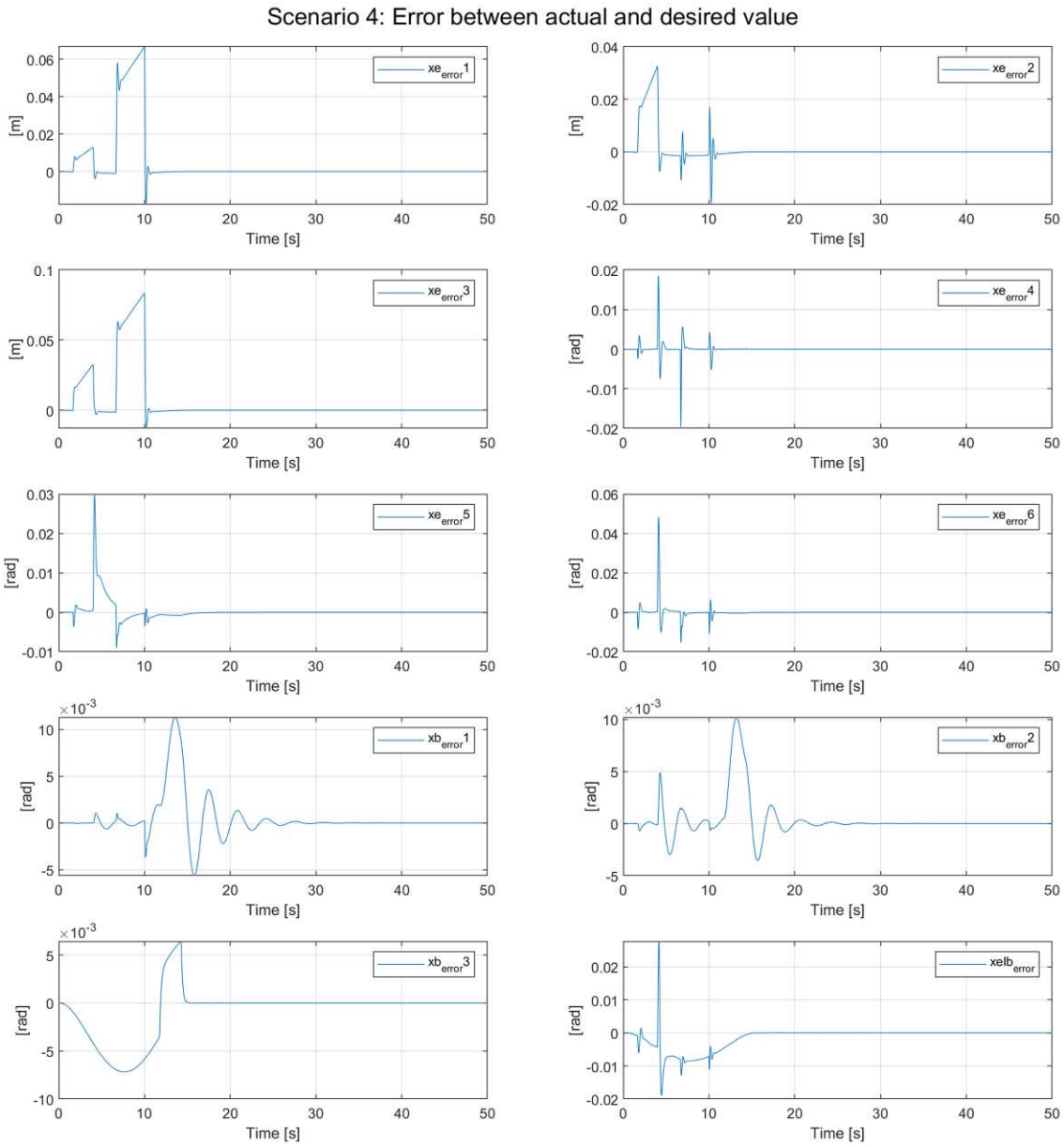


Figure 5.17: Scenario 4: Error of each parameter for Task 1 to 3

Use-Case 2:
Analogous, use-case 2 presents the results of the experiments using the feedback loop. Figure 5.18 visualizes the comparison of the trajectories, while Figure 5.19 shows the error between the curves. The controller demonstrates a remarkable ability to suppress external forces, and even with significant forces, the controller's behavior remains largely unaffected. In this scenario, all tasks also converge successfully.

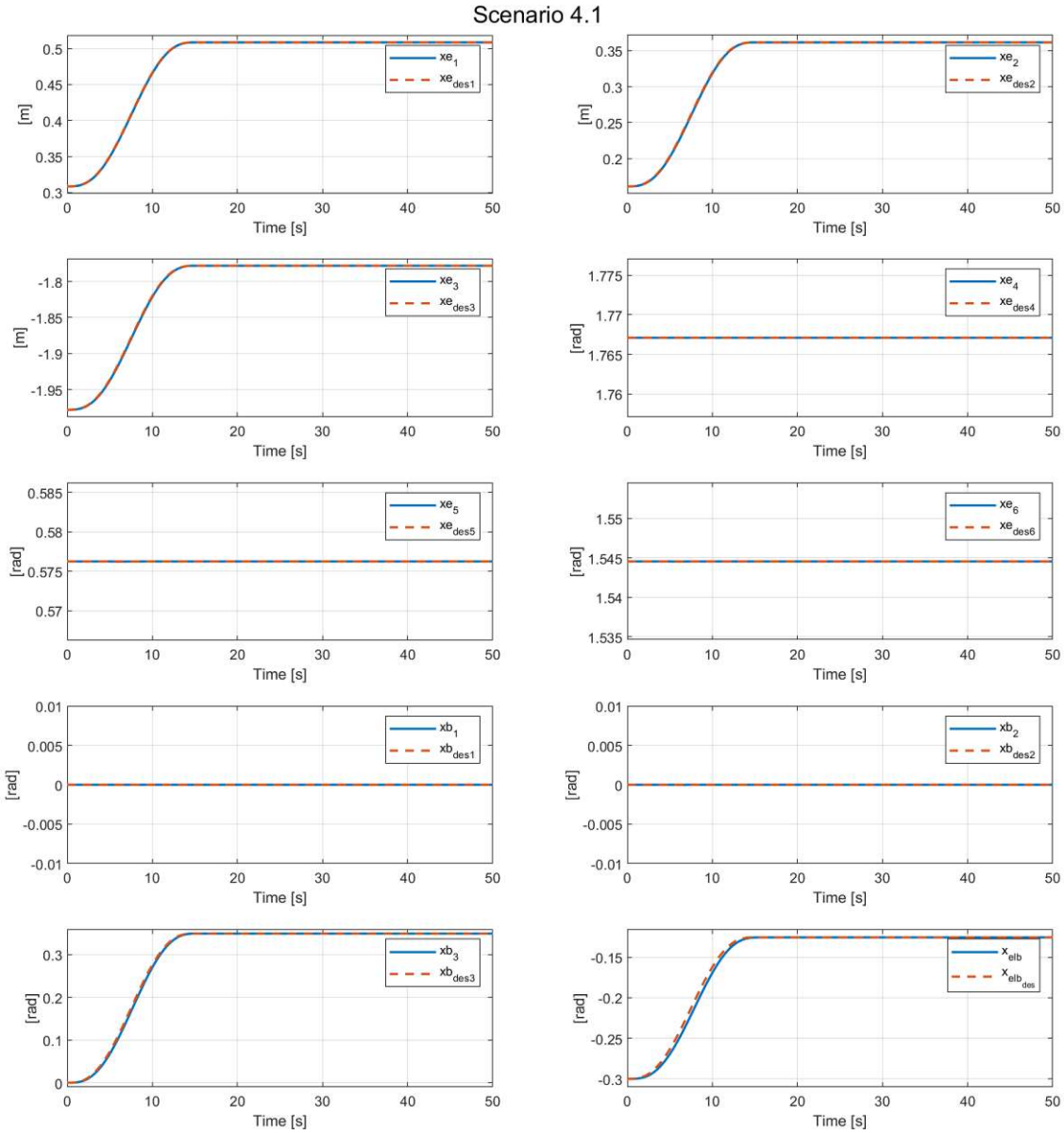


Figure 5.18: Scenario 4.1: Actual value compared with desired value for Task 1 to 3

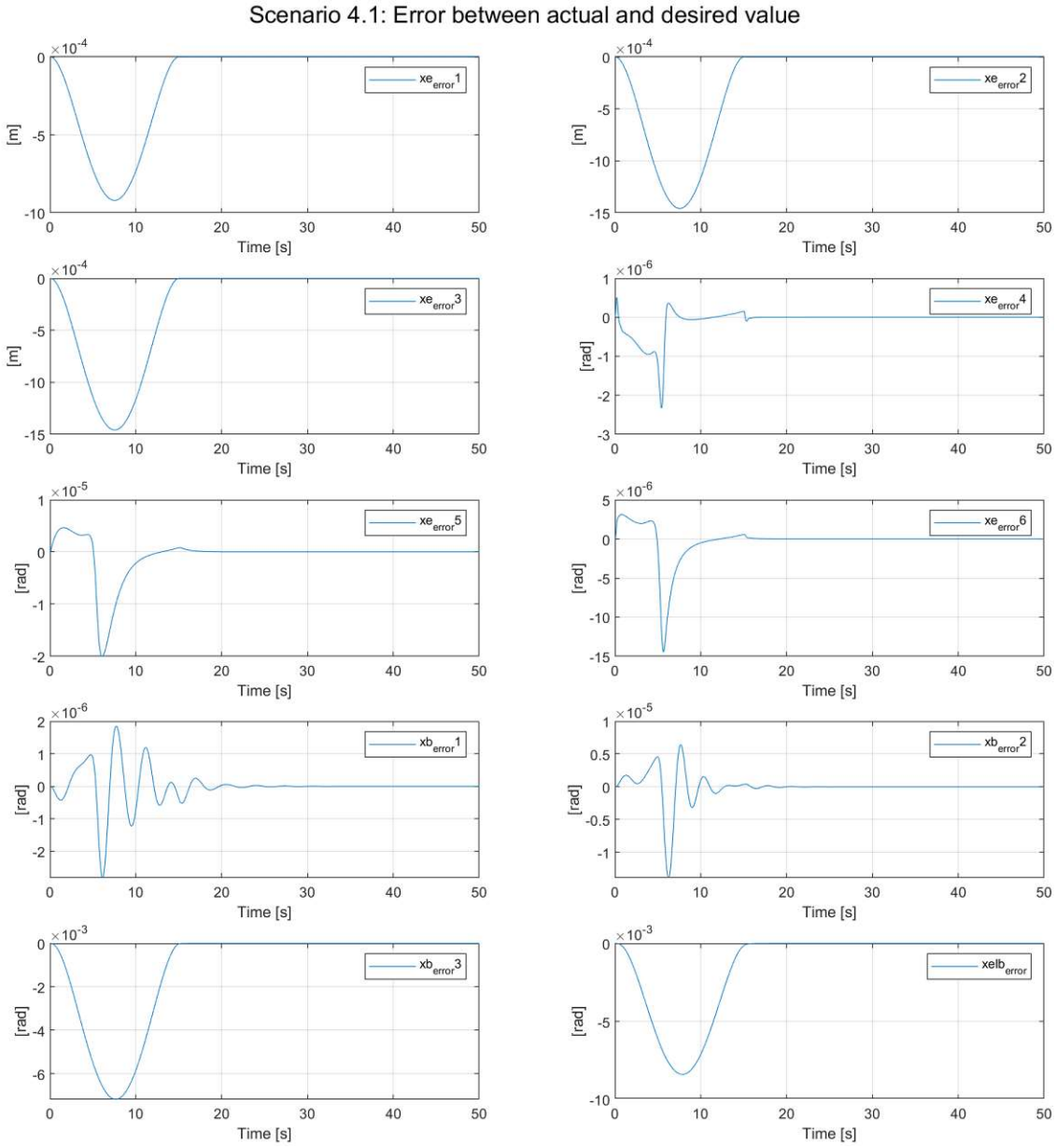


Figure 5.19: Scenario 4.1: Error of each parameter for Task 1 to 3

5.3.1 Discussion of the Hierarchical Controller results

Throughout all scenarios, the controller provides the required stability and speed of control. As a result, even in the event of external forces occurring, the controller is able to react quickly and efficiently to these influences and restore the desired trajectory. This indicates that the implemented controller has a fast adaptability to external disturbances.

The scenarios in which the feedback loop is activated, demonstrate effective suppression of external forces. The controller is able to successfully compensate for these forces and guide the robot endeffector along the desired path. The investigation of the scenarios with and without the feedback loop illustrate the significant impact of this feedback loop. Without the feedback of the current system states, the trajectories show certain deviations, which were quickly corrected by the controller once the external force disappeared. With the feedback loop activated, the controller is not only able to compensate for external forces, it also manages to guide the trajectories precisely and stably.

In scenario 4, various external forces and moments are simulated in all spatial directions. As shown by the results, the controller is able to guide the endeffector precisely even in the midst of these complex forces. The uncertainties regarding the magnitude of the external forces are successfully managed and in all scenarios performed, the system is observed to complete all tasks. This confirms the effectiveness of the controller in achieving the systemic goals despite external disturbances.

In conclusion, the implemented controller is suitable for multitask-control. The ability to suppress external forces and provide precise trajectory guidance makes it a promising approach for various applications in a dynamic and unknown environment.

5.4 Experimental Approach of the Optimizer

In the preceding Chapter 4, both the theoretical approach and the definition of the optimization problem are explained in detail. The purpose of this chapter is to focus on the practical implementation of this concept.

Based on the definition of the optimization problem (4.5) to (4.12), a non-linear static optimization problem is considered. It should be noted that for solving the optimization problem, the forward kinematics $f(q)$, the Jacobian matrix $J(q)$ and the gravity component $g(q)$ are evaluated in Maple. For this purpose and due to the calculation of the optimization using Matlab, the optimizer called *fmincon()* is chosen.

The function structure follows the syntax

$$x_{opt} = fmincon\Big(\text{fun}, x_0, A, b, A_{eq}, b_{eq}, lb, ub, \text{nonlcon}\Big) \in \mathbb{R}^{26\times 1} \tag{5.10}$$

$$\text{fun} = f_m(q_d, \tau, Fe) \tag{5.11}$$

$$x_0 = \{q_{init}, \tau_{init}, Fe_{init}\} \in \mathbb{R}^{26\times 1} \tag{5.12}$$

$$A = \mathbf{0} \tag{5.13}$$

$$b = \mathbf{0} \tag{5.14}$$

$$A_{eq} = \mathbf{0} \tag{5.15}$$

$$b_{eq} = \mathbf{0} \tag{5.16}$$

$$lb = \{q_{lb}, \tau_{lb}, Fe_{lb}\} \in \mathbb{R}^{26\times 1} \tag{5.17}$$

$$ub = \{q_{ub}, \tau_{ub}, Fe_{ub}\} \in \mathbb{R}^{26\times 1} \tag{5.18}$$

$$\text{nonlcon} = g_c(q_d, \tau, Fe) \tag{5.19}$$

The cost function (5.11) describes the target condition to be minimized, while (5.12) represents the initial seed value for the optimization. The parameters (5.17) and (5.18) represent the lower and upper limits of the variables. The constraints are defined within (5.19).

The reason to utilize (5.19) for constraints arises from their representation as a function, making them incompatible with the linear forms (5.13, 5.14, 5.15, 5.16).

As described in Section 4, the standard optimization problem is defined for minimizing the cost function, which also applies to (5.10). Hence, the transformation as described in equation (4.4) is implemented on the cost function (5.11), in order to maximize the given objective.

Adjusting the function (5.10) for the given problem, (5.11) is represented by (4.5) and the constraints (4.10), (4.11), (4.12) are declared in (5.19).

The initial condition x_0 , unless otherwise specified, denotes the joints of the system by q_{init}, defined in (A.1). The parameter $\tau_{init} \in \mathbb{R}^{10\times 1}$ is initialized as follows:

$$\tau_{init} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{5.20}$$

Similarly, $F_{init} \in \mathbb{R}^{6\times1}$ is set to:

$$F_{init} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{5.21}$$

The parameters for the limits (5.17) and (5.18), which include the constraints for the generalized coordinates (4.7), the torques (4.8) and the applied forces (4.9), can be found in Appendix A under Section A.3. The limits vary depending on which robot is selected for the overall system. Within this thesis, the system Franka Emika Panda is used.

Solving the optimization problem (5.10) of the aerial system symbolically leads to the output x_{opt}, including optimal generalized coordinates $q_{opt} \in \mathbb{R}^{10\times1}$, the optimal torques $\tau_{opt} \in \mathbb{R}^{10\times1}$ and the maximum reachable exerted endeffector force along all spatial dimensions $F_{opt} \in \mathbb{R}^{6\times1}$.

As previously discussed in Chapter 4, the optimizer is exclusively utilized offline, only considering the static case of the system.

5.4.1 Tests and evaluations of the Optimizer

The optimizer is evaluated through tests with various desired positions of the endeffector. The applied force, the pose of the robot and the values of the joint coordinates are thereby included. Based on these experiments, the hypothesis formulated in Chapter 4.3 is discussed and tested. The analysis will focus on the extent to which the results of the optimization tests can support or reject the assumptions and conclusions of the aforementioned hypothesis. While testing, the initial position of the endeffector is its home position with coordinates

$$x_{init} = \begin{bmatrix} 0.3085 & 0.1614 & -3.768 & 1.767 & 0.5762 & 1.545 \end{bmatrix}^T. \tag{5.22}$$

This leads to an exerted endeffector force F_h, which will serve as a reference for the conducted experiments.

$$F_h = \begin{bmatrix} 1.49 & 22.04 & 10.45 & 12.39 & 16.75 & 1.25 \end{bmatrix}^T [N] \tag{5.23}$$

The multirotor is replaced with a plate to achieve better visual effects, see Figure 5.20. Additionally, options for (5.10) are used, see

```
options = optimoptions('fmincon', 'MaxFunctionEvaluations', 3e+5,
    'MaxIterations', 2000, 'Algorithm', 'interior-point',
'EnableFeasibilityMode',true,'SubproblemAlgorithm','cg')}
```

The maximum possible evaluations and iterations are enlarged, additional, the algorithm is set to $interior-point$ for finding feasible points.

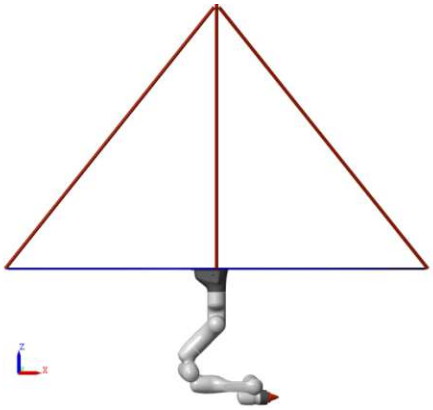


Figure 5.20: Initial Position of the robotic system

Scenario 1

Use-Case 1
While the initial position (5.22) of the endeffector remains fixed, Scenario 1 involves maximizing the force exerted at the endeffector in every arbitrary direction. This entails the optimizer striving to maximize the exerted force across all positions and orientations.

The result of the force vector in Newton demonstrates the locally maximized solution [1], representing the applied forces at the endeffector

$$F_{opt_{s1u1}} = \begin{bmatrix} 1.191 \cdot 10^{-8} & 55.32 & 58.32 & 70.12 & 119.9 & 12.05 \end{bmatrix}^T [N], \tag{5.24}$$

whose resulting motion can also be seen in image 5.21 (right) due to the optimized joint angles.

The results of the first scenario obtained by the optimizer for the joint angles $q_{opt_{s1u1}}$ in rad and torques $\tau_{opt_{s1u1}}$ in Nm are as follows:

$$q_{opt_{s1u1}} = \begin{bmatrix} 0.09 & 0.02 & 2.89 & 2.45 & -0.75 & 0.48 & -2.19 & -0.71 & 2.96 & 1.94 \end{bmatrix}^T$$

$$\tau_{opt_{s1u1}} = \begin{bmatrix} -87.00 & 87.00 & -87.00 & 76.02 & 82.76 & 87.00 & -87.00 & 12.00 & -12.00 & -12.00 \end{bmatrix}^T$$

Comparing the exerted force (5.23) of the endeffector before activating the optimizer with the results from the first scenario (5.24), it can be seen that expect for the exerted force along the x-axis, all other forces have increased. This observation may be attributed to the torques $\tau_{opt_{s1u1}}$ indicating the reach of the defined limits in nearly every torque of the robotic system, along with the limit reach of joint angle $q_{opt_{s1u1}}(3)$.

[1] This is indicated by the output of the optimizer $exitflag = 1$, which means that a local maximum has been found.

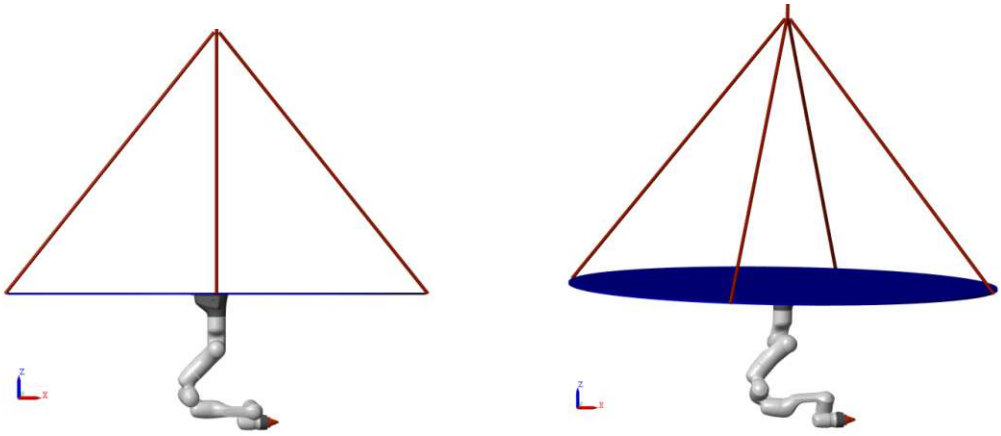


Figure 5.21: Scenario 1 Use-Case 1
Initial Pose (left), Optimized pose of robotic system (right)

Use-case 2
In use-case 2, instead of maximizing the applied force in all directions, the endeffector force F_d should be maximized only in one direction. For this purpose, the cost function in (4.5) is slightly adapted, see (5.25)

$$\text{maximize} \quad F_d \in \mathbb{R}^{1\times 1} \tag{5.25}$$

in order to maximize only the force in the desired direction. In addition, the constraint (4.10) is adjusted to (5.26)

$$g(q_d) = \tau + J(q_d)^T \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \cdot dir \cdot F_{dir}, \tag{5.26}$$

to consider only the direction of the exerted force with $dir \in \mathbb{R}^{6\times 1}$.
E.g. $dir = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ describes the specialized force consideration in x-direction.

Repeating use-case 1 with the adapted optimization problem, in which the force in x-direction is to be maximized, the resultant force vector gives the following values:

$$F_{opt_{s1u2}} = \begin{bmatrix} 78.05 & 14.52 & 15.03 & 14.34 & 14.45 & 14.41 \end{bmatrix}^T [N] \tag{5.27}$$

The joint angles $q_{opt_{s1u2}}$ and torques $\tau_{opt_{s1u2}}$ result as follows:

$$q_{opt_{s1u2}} = \begin{bmatrix} 0.21 & -0.01 & -1.38 & 2.59 & -0.71 & -2.1 & -0.89 & -2.23 & 1.35 & 2.13 \end{bmatrix}^T$$

$$\tau_{opt_{s1u2}} = \begin{bmatrix} -87.00 & -87.00 & -2.45 & 2.44 & -87.00 & -20.11 & -38.08 & 9.02 & 4.07 & 0.04 \end{bmatrix}^T$$

It is evident that F_x in (5.24), which was previously approximately zero Newton, has now been locally maximized [2] in (5.27), and the other values are no longer considered. The resulting position, as shown in Figure 5.22 (right), illustrates the inclination of the base. In this particular scenario, it is evident that gravity is not utilized. Instead, the pose of the robotic system is optimized for maximum force exertion.

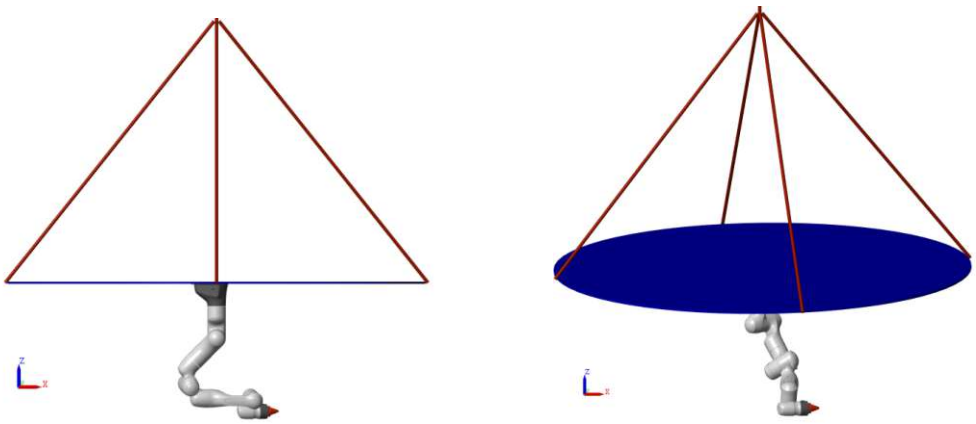


Figure 5.22: Scenario 1 Use-case 2
Initial Pose (left), Optimized pose of robotic system (right)

Scenario 2

In Scenario 2, the focus is on maximizing the force along the x-axis while simultaneously pursuing the goal of reaching point (5.28).

$$x_{des_{s2}} = \begin{bmatrix} 0 & 0 & -3.7 & 0 & 1.57 & 0 \end{bmatrix}^T \tag{5.28}$$

The result of the experiment performed shows that the resulting force vector $F_{opt_{s2}}$ is locally maximized [3] in the direction of the x-axis, as desired. This confirms the desired maximum of the force in this dimension, while the desired target pose retains its focus.

$$F_{opt_{s2}} = \begin{bmatrix} 234 & 9.418 & 9.418 & 9.418 & 9.418 & 9.418 \end{bmatrix}^T [N] \tag{5.29}$$

The optimization process yields the following joint angles $q_{opt_{s2}}$ and torques $\tau_{opt_{s2}}$:

$$q_{opt_{s2}} = \begin{bmatrix} 0.01 & 0.09 & -1.32 & 0.2 & -1.39 & -0.69 & -2.41 & -2.00 & 2.00 & 1.09 \end{bmatrix}^T$$

$$\tau_{opt_{s2}} = \begin{bmatrix} 25.57 & 71.97 & 1.76 & -1.73 & -4.32 & -87.00 & -87.00 & 12.00 & -12.00 & -0.03 \end{bmatrix}^T$$

[2] This is indicated by the output of the optimizer $exitflag = 1$, which means that a local maximum has been found.

[3] This is indicated by the output of the optimizer $exitflag = 1$, which means that a local maximum has been found.

Comparing the results from the second scenario with those from scenario 1, indicates more clearly that in the second scenario, gravity is used to achieve maximum force. This can be seen by the inclination of the base and the orientation of the endeffector, see Figure 5.23 (right). The findings obtained by the results suggest that the outcome of the optimizer strongly depends on the desired position of the endeffector as well as the configuration of the robot itself (considering the defined limits). Another conclusion can be drawn that although the optimizer does perform a maximization, it does not always take gravity into account.

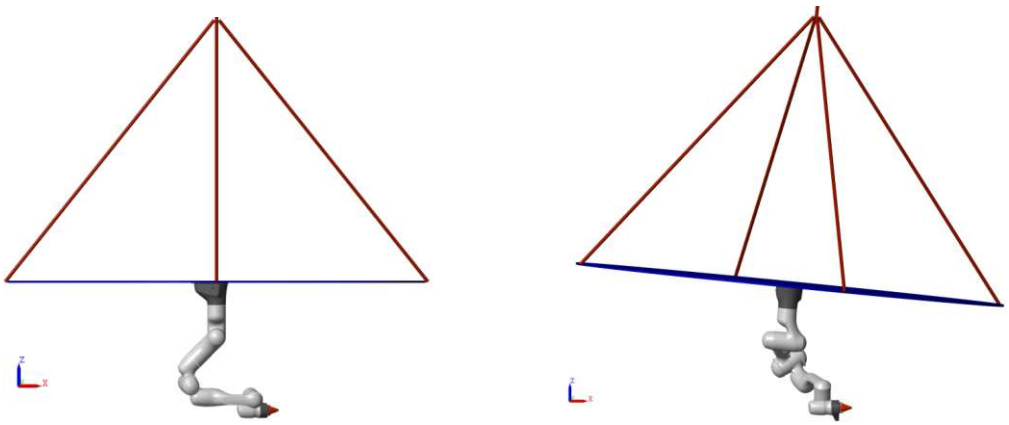


Figure 5.23: Scenario 2
Initial Pose (left), Optimized pose of robotic system (right)

Limits

In the final scenario, the limits of the optimizer are to be demonstrated. For this purpose, different positions along the x-,y- and z-axis are tested to analyze the behavior of the optimizer and to see a trend for which configurations gravity is used for maximizing the exerted force.

At the same time, the orientation of the endeffector (5.30) is specified and fixed with

$$x_{lim_{orient}} = \begin{bmatrix} 0 & 1.57 & 0 \end{bmatrix}^T. \tag{5.30}$$

Use-case 1
The approach initially starts by specifically maximizing the force applied along the x-axis, with the goal of determining the maximum tilt of the base. This is done by varying the pose of the endeffector in x- and z- direction. The focus remains on exploring the limits of what the robot can achieve while continuing to use gravity as a supporting element. This process aims to understand and define the robot's reach, the base tilt of the whole system and the performance under these conditions.

The experiment clarifies that it is possible to achieve precise poses within the ranges

$$\begin{aligned} x_{x,min} &= -0.1 \\ x_{x,max} &= 0.725 \\ x_{z,min} &= -3.6 \\ x_{z,max} &= -3.65 \end{aligned}$$

while $x_y = 0$ i.e. these are the limits for the determining the maximum possible tilt of the base.

The application of these limit values $x_{x,min}, x_{z,min}$ provide the following local [4] results for exerted force along the x-axis in N, the joint values in rad and the corresponding torques in Nm:

$$\begin{aligned} F_{opt_{x,min}} &= \begin{bmatrix} 186.2 & 9.275 & 9.275 & 9.275 & 9.275 & 9.275 \end{bmatrix}^T \\ q_{opt_{x,min}} &= \begin{bmatrix} -0.01 & 0.09 & 2.897 & 1.649 & -1.763 & 0.36 & -2.92 & 1.72 & 1.90 & -1.11 \end{bmatrix}^T \\ \tau_{opt_{x,min}} &= \begin{bmatrix} -28.29 & 87.00 & -1.14 & 1.13 & 1.37 & 31.59 & -87.00 & -12.00 & -9.27 & -0.03 \end{bmatrix}^T \end{aligned}$$

For the visual representation of the results, reference is given to Figure 5.24 (right). In order to illustrate a comparison when the tilt limit of the base is reached and the use of gravity is discontinued, Figure 5.25 is provided.

[4] This is indicated by the output of the optimizer $exitflag = 1$, which means that a local maximum has been found.

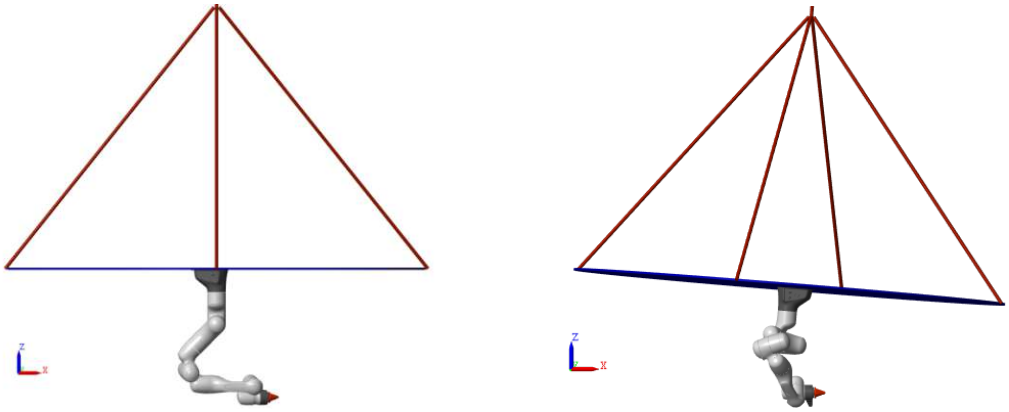


Figure 5.24: Scenario Limit for $x_{x,min}$
Initial Pose (left), Optimized pose of robotic system (right)

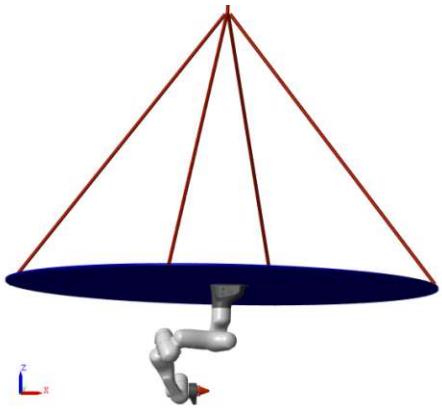


Figure 5.25: Optimizer result beyond limit

Applying the same procedures to the limits in the positive range of the x-axis results in the following values for the local maximized [5] exerted force along the x-axis in N, the joint values in rad and the corresponding torques in Nm:

$$F_{opt_{x,max}} = \begin{bmatrix} 72.81 & 10.14 & 10.14 & 10.14 & 10.15 & 10.15 \end{bmatrix}^T$$

$$q_{opt_{x,max}} = \begin{bmatrix} 0.00 & 0.01 & 0.00 & 0.59 & 1.05 & -1.34 & -0.84 & 1.93 & 2.83 & -0.87 \end{bmatrix}^T$$

$$\tau_{opt_{x,max}} = \begin{bmatrix} -8.09 & -87.00 & 0.05 & -0.05 & 87.00 & 41.05 & -26.03 & 3.56 & -5.42 & -0.03 \end{bmatrix}^T$$

Displaying the results graphically shows the pose of the system with the inclination of the base in Figure (right). The tilt of the base is very small, which is reflected in the joint coordinates.

[5] This is indicated by the output of the optimizer $exitflag = 1$, which means that a local maximum has been found.

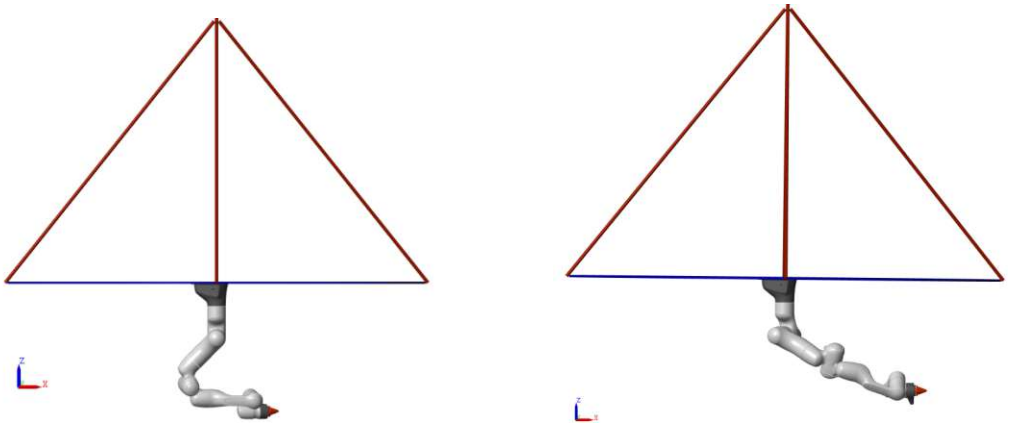


Figure 5.26: Scenario Limit for $x_{x,max}$
Initial Pose (left), Optimized pose of robotic system (right)

Beyond the boundary involving the variation of the z-component, Figure 5.27 shows that gravity is not considered in the optimization. This is evident from the tilt of the base, which does not incline in the same direction as the tip of the endeffector.

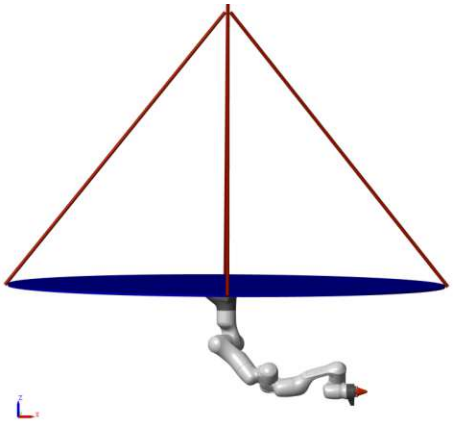


Figure 5.27: Optimizer result beyond limit

Use-Case 2
The procedure outlined in Use-case 1 is replicated, with the objective now shifted towards maximizing force along the y-axis. This adjustment involves the application of different positional values to the endeffector in the y- and z- directions. Thereby, (5.30) is adapted to

$$x_{lim_{orient}} = \begin{bmatrix} 1.57 & 1.57 & 0 \end{bmatrix}^T \tag{5.31}$$

and remains constant throughout the process.

Boundaries for the inclination of the base, which are identified in the context of force

maximization along the y-axis, can be described by the following values:

$$x_{y,min} = -0.1$$
$$x_{y,max} = 0.725$$
$$x_{z,min} = -3.8$$
$$x_{z,max} = -3.6$$

During the experimental procedure, it is evident that $x_x = 0$.

By identifying the minimum values, i.e. the limit values for the optimizer, as summarized in $x_{y,min}$ and $x_{z,min}$, a locally [6] maximum force is achieved along the y-axis. This characteristic leads to specific joint angles in $q_{opt_{y,min}}$ and the corresponding torques in $\tau_{opt_{y,min}}$.

$$F_{opt_{y,min}} = \begin{bmatrix} 2.04 & 143.4 & 3.09 & 3.36 & 3.63 & 2.51 \end{bmatrix}^T$$

$$q_{opt_{y,min}} = \begin{bmatrix} -0.11 & 0.00 & 2.897 & -2.897 & -1.62 & -0.31 & -2.23 & -2.08 & 1.41 & 0.73 \end{bmatrix}^T$$

$$\tau_{opt_{y,min}} = \begin{bmatrix} -87.00 & 13.59 & 1.69 & -1.72 & -39.75 & -36.58 & -34.96 & 12.00 & -12.00 & -0.03 \end{bmatrix}^T$$

The visualization is depicted in Figure 5.28.

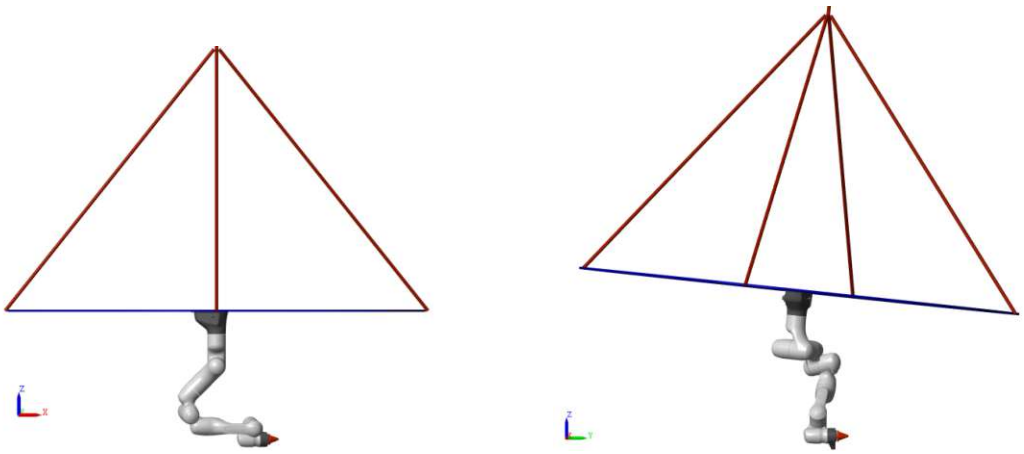


Figure 5.28: Scenario Limit for $x_{y,min}$
Initial Pose (left), Optimized pose of robotic system (right)

Violating the limits by further reducing the x_y value leads to a pose as in Figure 5.29.

[6] This is indicated by the output of the optimizer $exitflag = 1$, which means that a local maximum has been found.

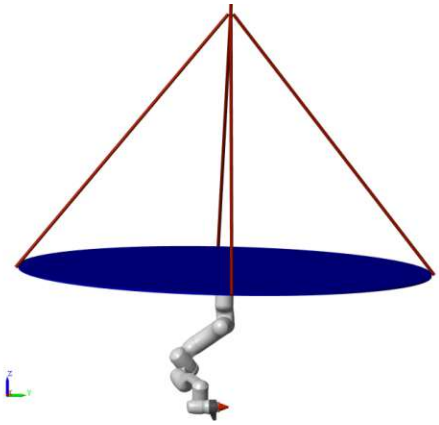


Figure 5.29: Optimizer result beyond limit

By achieving the limits for the positive y-direction, it is possible to obtain the results for the maximum global [7] force exerted by the endeffector in N, along with the corresponding joint angles in rad and the torques in Nm.

$$F_{opt_{y,max}} = \begin{bmatrix} 7.15 & 74.64 & 7.15 & 7.15 & 7.15 & 7.15 \end{bmatrix}^T$$

$$q_{opt_{y,max}} = \begin{bmatrix} -0.01 & 00 & 1.57 & 0.64 & 1.13 & -1.41 & -0.92 & 2.20 & 2.81 & -1.05 \end{bmatrix}^T$$

$$\tau_{opt_{y,max}} = \begin{bmatrix} 87.00 & -10.74 & -0.22 & 0.22 & 87.00 & 47.41 & -22.83 & 4.01 & -4.23 & -0.03 \end{bmatrix}^T$$

The graphical representation of the optimization results is shown in Figure 5.30 (right).

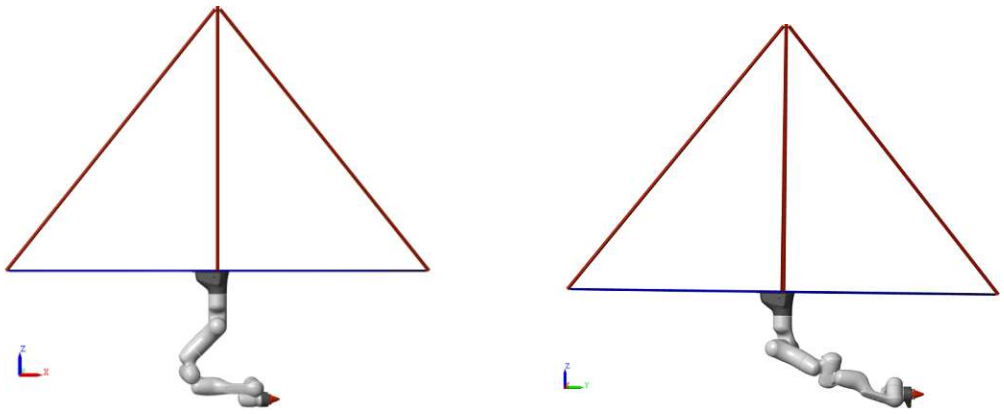


Figure 5.30: Scenario Limit for $x_{y,max}$
Initial Pose (left), Optimized pose of robotic system (right)

[7] This is indicated by the output of the optimizer $exitflag = 2$, which means that a global maximum has been found.

Exceeding the specified limits is shown in Figure 5.31 by increasing the y-position of the endeffector. It is apparent that gravity is no longer used to maximize the force.

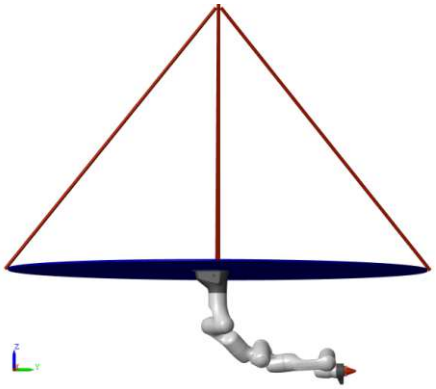


Figure 5.31: Optimizer result beyond limit

5.4.2 Discussion of the Optimization Results

The first scenario illustrates that from the starting position of the endeffector, capable of exerting a force (5.23), the optimizer can effectively maximize this force either in all spatial directions or in a specified direction. Scenario 2 demonstrates the application of the optimizer in a different configuration.

Both scenarios illustrate that the optimizer can maximize the force by adjusting the configuration of the robot. However, they also demonstrate that, contrary to the assumption in the hypothesis where the optimizer would use gravity to maximize the force, gravity is not always used.

This can be seen in Figure 5.21 Scenario 1 Use-case 1, where it becomes clear that gravity is not used based on the tilt of the base. This might be due to the output of the optimizer, in particular the generalized coordinates and the torques, which are very close to the specified limits of the robot and therefore the use of gravity may not be possible. Another reason could be that maximizing the force without gravity, i.e. using the torques of the system, might lead to better results in terms of exerted output force.

In contrast, scenario 2 shows that the optimizer is less restricted by the limits, which could be the reason for the increase in the maximized force. It is also apparent, that gravity is used in this case.

Important findings are therefore that the optimizer maximizes the force at the endeffector, but the maximum force is not always associated with the use of gravity. This may imply that better results can be achieved without using gravity. The results of the experiments also show, that the outcome of the optimizer is strongly dependent on the selected desired pose and on the configuration of the robot itself.

The limit scenarios of the experiments should indicate the limits of the optimizer, i.e. the pose of the robot-endeffector is changed to find the maximum possible tilt of the base. Comparing the assumptions in the hypothesis, it is assumed that the more the base is tilted, the more the robot arm is stretched and gravity is used for maximizing the exerted force.

Therefore, the experiments describe the desired positions up to which the base is tilted in such a way that gravity is used. The beyond-limit Figures 5.25, 5.27, 5.29 and 5.31 show, that outwith the boundaries, the base is not tilted in the same direction as the direction of the robot tip. It can also be seen that in these examples, the base is less tilted and the force at the endeffector is taken from the robot arm itself.

In summary, this section provides evidence of the functionality of the optimizer. It also indicates that the results of the optimizer are strongly influenced by the configuration of the robot and the desired pose. Furthermore, the results show that the optimizer does not always maximize the force based on the use of gravity, but rather achieves results through the torque of the arm. This means that even within the limits identified, a configuration which maximizes the force but does not use gravity, is possible.

5.5 Use-Case Implementation and Evaluation

The purpose of this section of the thesis is to describe a practical use-case for the robotic system and test it in simulation. This involves the use of a combination of the HC and the optimizer, which were examined in detail in the previous chapters with regard to their performance and functionality. The focus is now on combining these two components in a shared application in order to fulfill a precisely defined use-case in the best possible way. This integrated approach enables an efficient implementation of the tasks set, with the preceeding individual tests serving as a basis to ensure that the combined operation of both elements works efficiently.

Thus, this chapter is dedicated to provide a detailed definition of the use-case and an in-depth discussion of the simulation results. The goal is to achieve a deeper understanding of the system's performance and challenges.

5.5.1 Use-Case Definition

The idea of the use-case is to use the crane to move the SAM to a specific position, from where the system navigates to a target object and maximizes the force exerted on the endeffector when it encounters the object. The sequence of this use-case performed by the robotic system is divided into several steps.

Step 0: Initially, the robot starts in its home position, see Figure 5.1, defined initial joint values in (A.1) and corresponding derivatives in (A.2), with the HC deactivated.

Step 1: In the first step, the crane moves the system to the desired position in order to get closer to the object to be touched. As the HC is not enabled during this crane movement, the system is stabilized in its initial position using a PD-control with gravitation compensation in the joint-space. The inertia of the system becomes apparent during the crane movement, which causes the system to oscillate when it reaches the crane position.

Step 2: Following a few seconds, the HC is activated to dampen the oscillation. At the same time, it controls the movement of the endeffector to reach the object.

Step 3: When the object is reached, the optimizer is activated to maximize the force exerted on the endeffector (along the x-axis).

5.5.2 Practical Implementation

The practical and simulative implementation follows the steps as defined in Chapter 5.5.1.

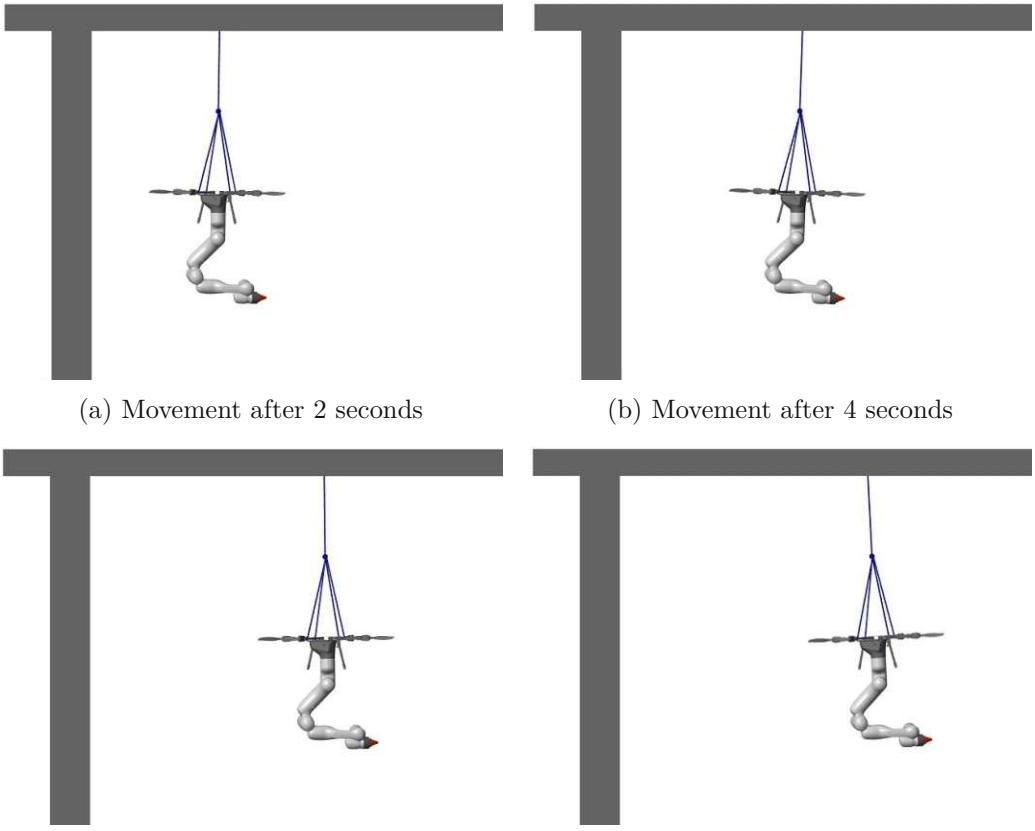
Step 0 and 1
Starting from the initial position, the entire system is displaced by 1.05 meter in x-direction using the crane. Thereby, the trajectory movement follows a 5^{th} order polynomial function. The velocity of motion of the system equals to 0.25 meters/second. During

this time, the PD regulator with gravitation compensation is enabled, operating in the joint-space of the system and precisely stabilizing and controlling it in its initial position. The control formulation $\tau_{PD} \in \mathbb{R}^{10\times1}$ for the controller is given by

$$\tau_{PD} = g(q) - K_{PD} \cdot (q - q_{init}) - D_{PD} \cdot dq \tag{5.32}$$

for which $g(q) \in \mathbb{R}^{10\times1}$ describes the gravitational components and $K_{PD} \in \mathbb{R}^{10\times10}$ and $D_{PD} \in \mathbb{R}^{10\times10}$ are positive definite and symmetric controller matrices, defined in Table A.7.

A time sequence of this movement during the simulation is shown in the following Figure 5.32. These figures illustrate both, the actual movement and the inertia of the system during this process.



(a) Movement after 2 seconds (b) Movement after 4 seconds

(c) Movement after 6 seconds (d) Movement after 8 seconds

Figure 5.32: Time sequence of crane movement

Step 2
Considering the current location of the robotic system, the HC is activated after a period of 9 seconds (whereby the first 8 seconds are required by the crane). The tasks to be

performed by the controller are configured as described below:

$$x_{1,des} = \begin{bmatrix} 0.53 & 0 & -2 & 1.767 & 0.5762 & 1.545 \end{bmatrix}^T \in \mathbb{R}^{6\times 1}$$
$$x_{2,des} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \text{deg} \in \mathbb{R}^{3\times 1}$$
$$x_{3,des} = x_{elb,init} \text{ deg} \in \mathbb{R}^{1\times 1}$$

Before activating the optimizer, it is first checked whether the closed-loop system is continuing to be stable. While convergence has been demonstrated in all experiments where the HC is examined, it is worth noting that the external force is well known in these experiments. However, in the present simulation, the external disturbance, representing the dynamics of the crane, remains unknown. In other words, the crane dynamics is unknown and acts on the SAM as an external disturbance.

To enhance the responsiveness of the system to external disturbances and reduce its sensitivity, adjustments are made by increasing the stiffness and damping characteristics of the system. The parameters are modified in accordance with Table 5.1.

Table 5.1: New Stiffness and Damping Coefficients of Hierarchical Controller

Index	Stiffness K	Damping D
1	4919	140.3
	1961	88.57
	1957	88.47
	4.66	4.32
	144.7	24.06
	691.8	52.61
2	3339	115.6
	1443	75.96
	736.5	54.28
3	133.5	23.11

Figures 5.33 and 5.34 illustrate the execution of the desired trajectories by the system. Besides showing the sequential progression of the robot through its steps (Step 0, 1 and 2) along the prescribed path, it also exemplifies convergence towards the desired outcome. This signifies that the system maintains stability, enabling the progression to Step 3.

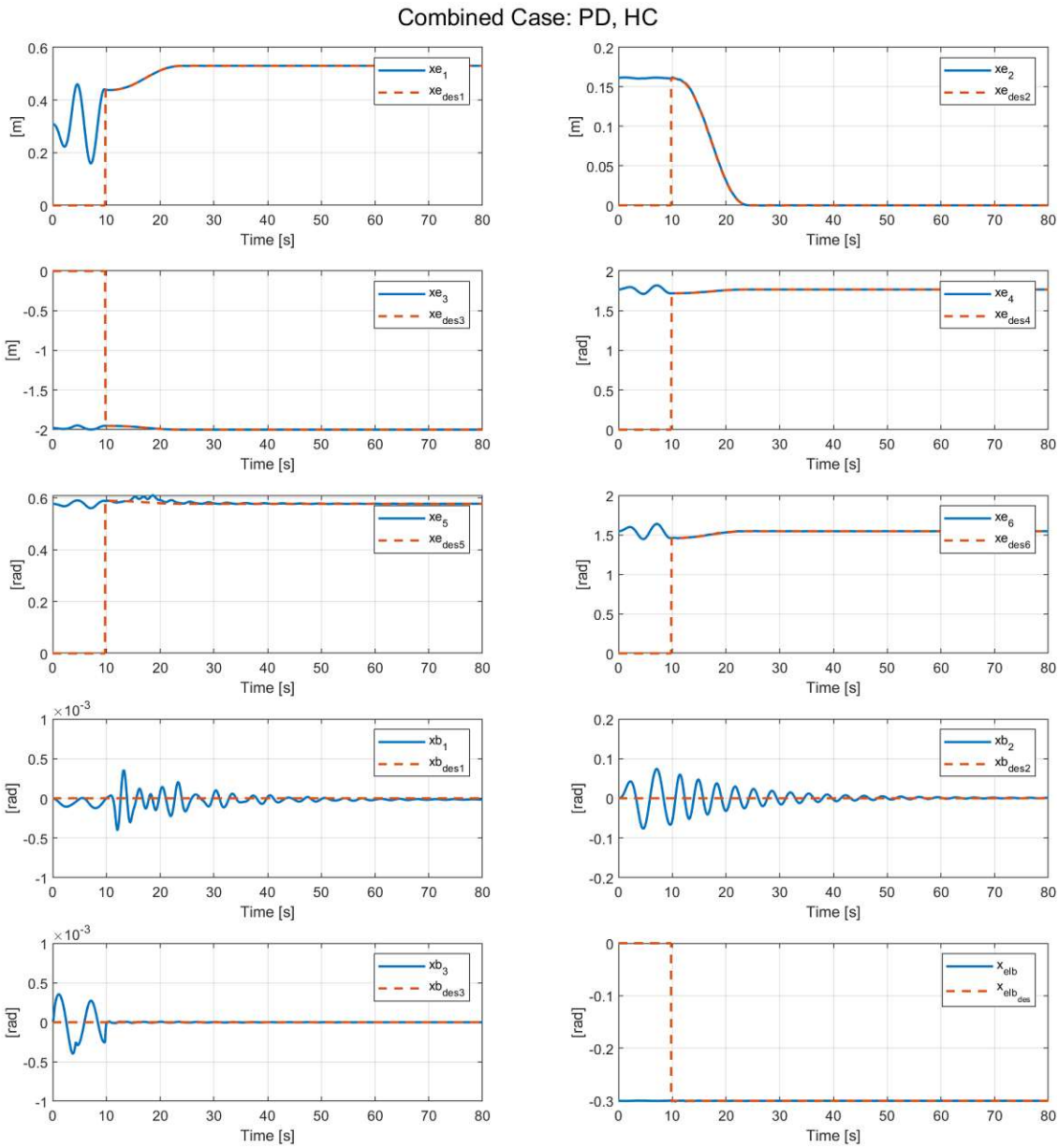


Figure 5.33: Use-Case without optimizer: Actual value compared with desired value for Task 1 to 3

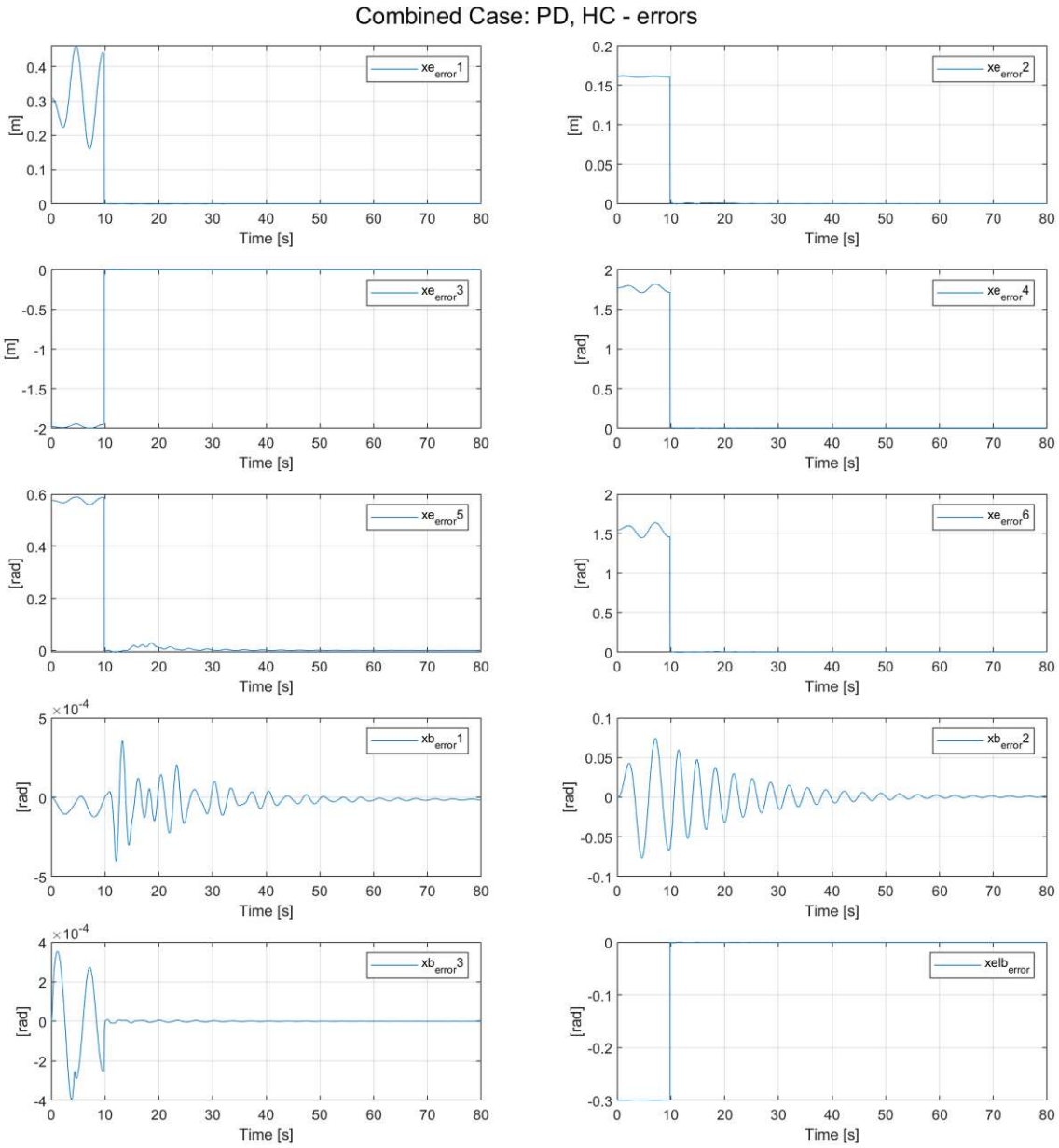


Figure 5.34: Use-Case without optimizer: Error of each parameter for Task 1 to 3

Figure 5.35 and 5.36 indicates the termination of the external error more precisely by displaying the error with the start of the activation of the HC controller.

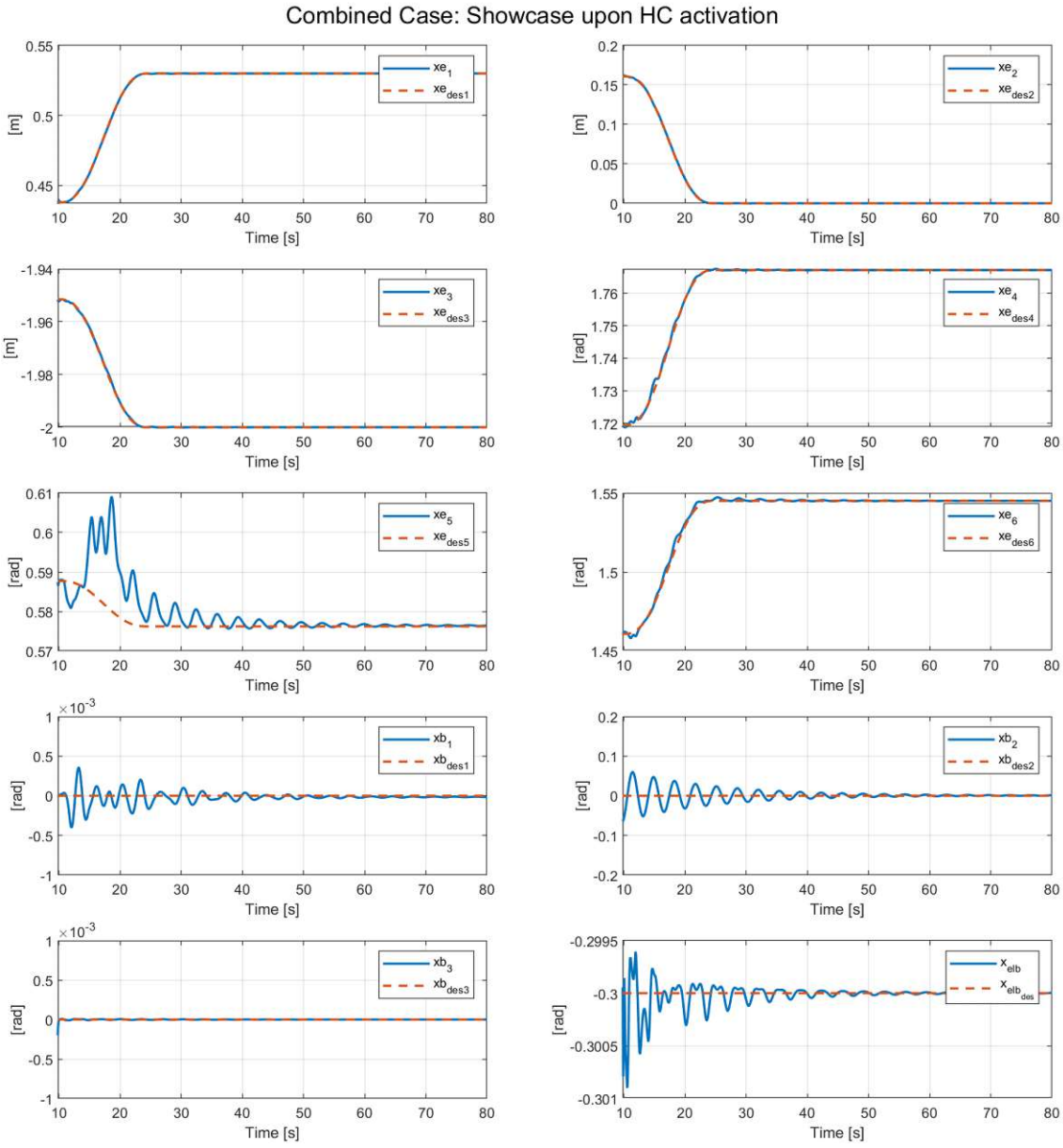


Figure 5.35: Use-Case without optimizer: Showcase upon HC activation

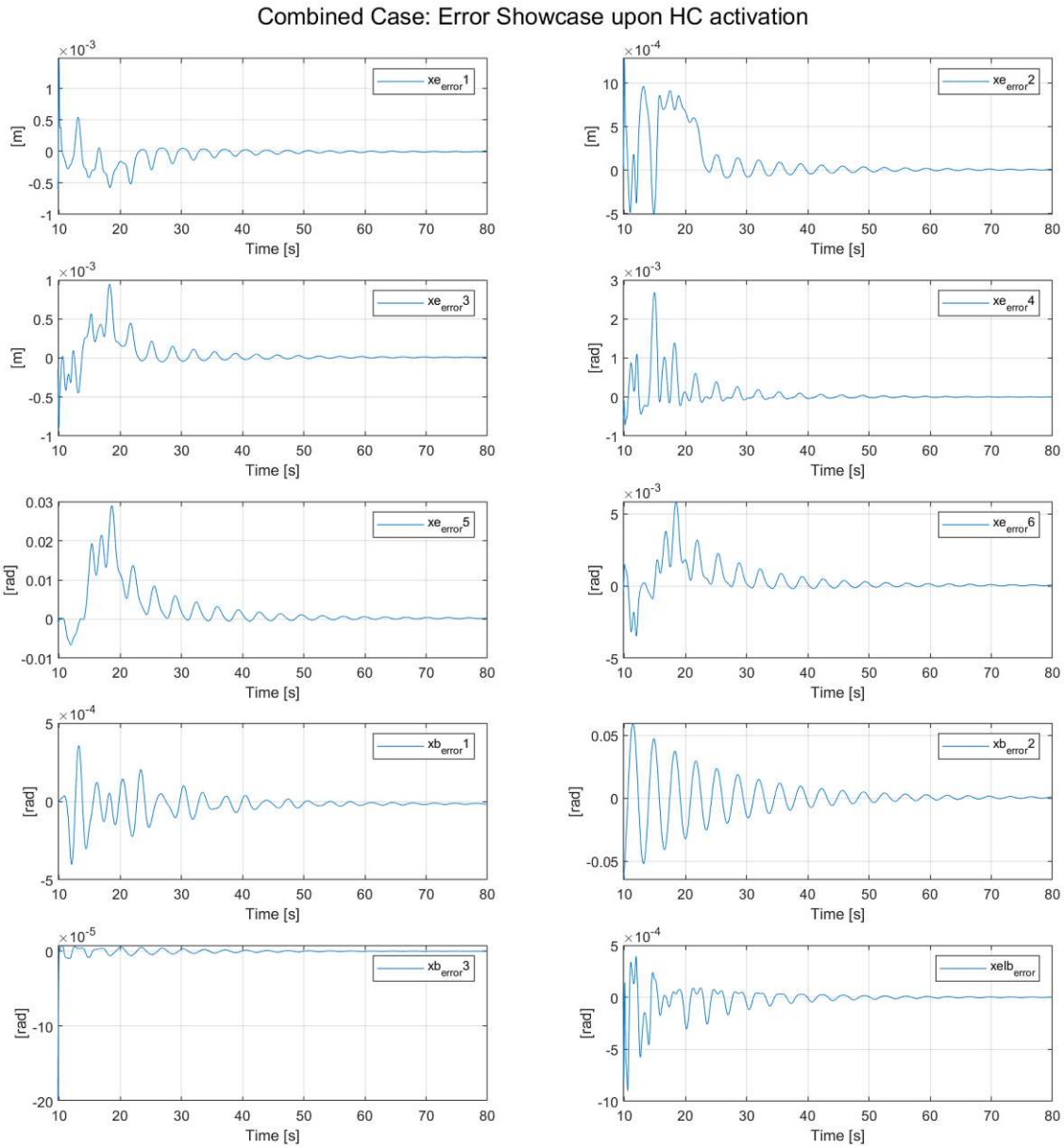


Figure 5.36: Use-Case without optimizer: Error Showcase upon HC activation

Step 3
The robotic system follows the specified trajectory up to the point where the endeffector touches the target object, which is the red ball, i.e. the top of a flag pole (see Figure 5.37). When contact occurs between the endeffector and the object, the optimizer is

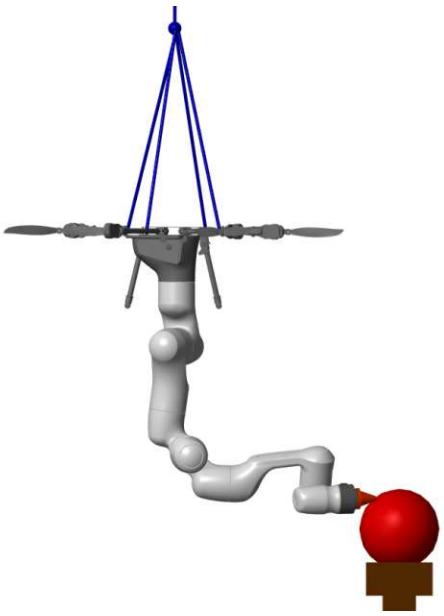


Figure 5.37: Use-Case: Step 3

activated and maximizes the force along the x-axis to

$$F_{Opt} = \begin{bmatrix} 58.56 & 23.05 & 20.75 & 22.48 & 20.83 & 24.07 \end{bmatrix}^T [N]. \tag{5.33}$$

5.5.3 Discussion

The detailed analysis of Images 5.33, 5.34 and 5.36 provide a comprehensive insight into the step-by-step Use-case execution and reveals a clear sequence of control elements. At the beginning, the PD controller dominates for the first 9 seconds, acting in the joint-space and aiming to stabilize the desired joint angles. This marks the initial control step.

In second 9, the seamless transition to the activation of the HC takes place, which can be recognized by the noticeable jump in the desired trajectory. The controller successfully demonstrates the following of the trajectory in the task-space.

As previously stated, the crane dynamics is considered as an unknown external influence affecting the entire system. Consequently, the crane-motion is not situated within the null space of Task 1. This is evident in the extended oscillation of the endeffector (see Figure 5.35) before convergence, attributable to the external crane dynamics - a phenomenon

that does not occur in the HC experiments. Since Task 2 is located in the null-space of Task 1 and the protracted convergence time of Task 1, this hierarchical interdependence is reflected in the convergence duration of Task 2. The same principle applies to Task 3.

Integrating Step 3 into the simulation, i.e. activating the optimizer at given contact, results in the maximization of the exerted force by the endeffector onto the touched object. The locally [8] maximized result is available in (5.33).

In summary, it is evident that the controller and optimizer can be effectively executed together. However, a significant challenge of this use-case arises from the unknown dynamics of the crane. Therefore, adjustments to the stiffness and damping parameters are necessary to maintain stability of the closed-loop system.

[8] This is indicated by the output of the optimizer $exitflag = 1$, which means that a local maximum has been found.

6 Conclusion and Outlook

This thesis is dedicated to the multitask control of a suspended aerial manipulator. The system consists of 10 degrees of freedom and is controlled using a whole body controller. The available degrees of freedom are used accomplishing a total of 3 tasks, which can be executed in a hierarchical fashion. The entire system operates in unknown environments. When an object is touched, the force exerted by the endeffector is maximized. This requires an additional optimization process.

The control structure is based on a hierarchical approach and the tasks are assigned with different priorities. During operation, the task with the highest priority is executed ensuring that tasks with a lower priority have no influence or modification. This is an outstanding advantage of the overall design, since the hierarchical structure enables effective decoupling of the different jobs, allowing tasks with a lower priority to operate in the null-space of the higher-priority tasks. Consequently, the non-convergence of a lower-priority assignment has no effect or impact on higher priority tasks. Similarly, the optimizer is designed in such a way to prevent modifications with respect to Task 1. The desired behavior is confirmed by an intensive investigation with simulative experiments.

The developed framework provides the possibility to easily adapt the program for different robotic configurations, parameters and tasks without requiring in-depth understanding of the code. It thus represents a flexible basis for the implementation and testing of multitask control systems for suspended aerial manipulators. By specifying the individual tasks in task-space, their description is simple and intuitive.

Despite the many advantages of this approaches, there are also challenges. Due to the complex structure of the hierarchy within the control system, numerous transformations between joint-, decoupled- and task-space are required. In addition, the robot must fulfill certain properties, such as its redundancy.

The results of the tasks in the decoupled task-space are not intuitive, which requires additional transformations. Due to the arising computational costs, performing simulations thus requires a powerful computer structure. Given the complexity of the overall system, the functionality of the optimizer is impaired by the numerous constraints. However, it is difficult to relax the constraints, as some of these are predefined by the kinematics of the robot itself.

Considering the achieved progress and the identified challenges, the thesis offers a promising starting point for future developments and optimizations:

- So far, the optimizer has only been used in the static case, i.e. the speed and acceleration of the system equals to zero. It would be interesting to design an online variant of the optimizer. The integration for the dynamic case would represent a significant advance in the development of multitask control. This extension could enable continuous, real-time adaptation to the changing dynamics of the system, taking into account velocity and acceleration data and ensuring adaptive control in dynamic environments.
- In terms of optimizing energy efficiency, an in-depth study could aim to improve the energy efficiency of the system. This could be done by developing strategies to minimize energy consumption, especially in situations with high redundancy or low load. In such scenarios, the robot benefits from more degrees of freedom and could therefore select more energy-efficient movements to complete the task.
- To ensure the robustness and applicability of the suspended aerial manipulator in real-life scenarios (rather than simulative), a comprehensive series of tests in different environments and working conditions would be beneficial. This could enable the identification and resolution of challenges in real-life scenarios.

Addressing the challenges for the future could not only improve the efficiency and performance of the presented control system, but also improve its applicability to a wider range of real-life scenarios.

A System parameters

A.1 Default generalized coordinates

The initial values for the generalized coordinates have been defined in $[rad]$ as follows

$$q_{init} = \begin{bmatrix} 0 & 0 & 0 & 0 & -0.7854 & -0.3 & -2.3562 & 0 & \pi & 0.7854 \end{bmatrix}^T \tag{A.1}$$

whereas the initial conditions for the corresponding derivations are defined in $[rad/sec]$ as

$$dq_{init} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{A.2}$$

A.2 Hierarchical controller parameters

The empirically established values of K_i and D_i are listed in Table A.1. In this context, the index i corresponds to the priority of the task, i.e. $i = 1$ is therefore the task with the highest priority, described in (5.1). It is apparent that the damping coefficient is defined as $D_i = 2\sqrt{K_i}$.

Table A.1: Stiffness and Damping Coefficients of Hierarchical Controller

Index	Stiffness K	Damping D
1	2951	108.6
	1177	68.61
	1174	68.53
	2.79	3.34
	28.95	10.76
	138.4	23.53
2	133.6	23.11
	144.3	24.02
	147.3	24.27
3	26.7	10.34

A.3 Optimizer Constraints due to given robotic system

The upper and lower limits for the generalized coordinates of the robot Franka Panda Emika are listed in Table A.2.

Table A.2: Generalized coordinates limits for Franka Emika Panda

Limits in rad	q_1	q_2	q_3	q_4	q_5
Minimum	-2.8973	-2.8973	-2.8973	-2.8973	-1.7628
Maximum	2.8973	2.8973	2.8973	2.8973	1.7628

Limits in rad	q_6	q_7	q_8	q_9	q_{10}
Minimum	-2.8973	-3.0718	-2.8973	-0.0175	-2.8973
Maximum	2.8973	-0.0698	2.8973	3.7525	2.8973

The upper and lower limits for the torque of the robot Franka Panda Emika are listed in Table A.5. The first three torque limits refer to the propellers mounted on the aerial platform.

Table A.5: Torque limits for Franka Emika Panda

Limits in Nm	τ_{prop_1}	τ_{prop_2}	τ_{prop_3}	τ_4	τ_5	τ_6	τ_7	τ_8	τ_9	τ_{10}
Minimum	-87	-87	-87	-87	-87	-87	-87	-12	-12	-12
Maximum	87	87	87	87	87	87	87	12	12	12

The upper and lower limits for the forces of the robot Franka Panda Emika are listed in Table A.6.

Table A.6: Exerted Endeffector force limits

Limits in N	F_x	F_y	F_z	F_φ	F_θ	F_ψ
Minimum	0	0	0	0	0	0
Maximum	∞	∞	∞	∞	∞	∞

A.4 PD controller with gravity compensation parameters

Using the matrices $K_{PD} \in \mathbb{R}^{10\times10}$ and $D_{PD} \in \mathbb{R}^{10\times10}$, which are defined for the PD controller and listed in Table A.7, specify the weighting of the joint coordinates and their derivatives in the joint space. Here K_{PD} represents the stiffness, while D_{PD} represents the damping.

Table A.7: Stiffness and Damping Coefficients of PD control

Joint Angle	Stiffness K	Damping D
q_1	500	31.3050
q_2	500	31.3050
q_3	500	31.3050
q_4	500	31.3050
q_5	500	31.3050
q_6	500	31.3050
q_7	50	9.8995
q_8	50	9.8995
q_9	50	9.8995
q_{10}	10	4.4272

Bibliography

[1] M. Orsag, C. Korpela, P. Oh, S. Bogdan, and A. Ollero, *Aerial manipulation*. Springer, 2018.

[2] F. Ruggiero, V. Lippiello, and A. Ollero, "Introduction to the special issue on aerial manipulation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2734–2737, 2018.

[3] I. Palunko, P. Cruz, and R. Fierro, "Agile load transportation : Safe and efficient load manipulation with aerial robots," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 69–79, 2012.

[4] C. M. Korpela, T. W. Danko, and P. Y. Oh, "Designing a system for mobile manipulation from an unmanned aerial vehicle," in *2011 IEEE Conference on Technologies for Practical Robot Applications*, 2011, pp. 109–114.

[5] S. Hamaza, I. Georgilas, G. Heredia, A. Ollero, and T. Richardson, "Design, modeling, and control of an aerial manipulator for placement and retrieval of sensors in the environment," *Journal of Field Robotics*, vol. 37, no. 7, pp. 1224–1245, 2020. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21963`. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21963`.

[6] Y. S. Sarkisov *et al.*, "Development of sam: Cable-suspended aerial manipulator," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5323–5329.

[7] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.

[8] J. Everaerts *et al.*, "The use of unmanned aerial vehicles (uavs) for remote sensing and mapping," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, no. 2008, pp. 1187–1192, 2008.

[9] F. Huber *et al.*, "First analysis and experiments in aerial manipulation using fully actuated redundant robot arm," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3452–3457.

[10] A. Suarez, G. Heredia, and A. Ollero, "Lightweight compliant arm with compliant finger for aerial manipulation and inspection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4449–4454.

[11] A. Yiğit, M. A. Perozo, L. Cuvillon, S. Durand, and J. Gangloff, "Novel omnidirectional aerial manipulator with elastic suspension: Dynamic control and experimental performance assessment," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 612–619, 2021.

[12] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[13] A. Dietrich, *Whole-body impedance control of wheeled humanoid robots*. Springer, 2016, vol. 116.

[14] A. Dietrich and C. Ott, "Hierarchical impedance-based tracking control of kinematically redundant robots," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 204–221, 2020.

[15] L. S. Mello, G. V. Raffo, and B. V. Adorno, "Robust whole-body control of an unmanned aerial manipulator," in *2016 European Control Conference (ECC)*, 2016, pp. 702–707.

[16] J. H. Park, "Compliance/impedance control strategy for humanoids," in *Humanoid Robotics: A Reference*, A. Goswami and P. Vadakkepat, Eds., Dordrecht: Springer Netherlands, 2016, pp. 1–20, ISBN: 978-94-007-7194-9. [Online]. Available: `https://doi.org/10.1007/978-94-007-7194-9_148-1`.

[17] N. Hogan, "Impedance control: An approach to manipulation," in *1984 American Control Conference*, 1984, pp. 304–313.

[18] C. Gabellieri, Y. S. Sarkisov, A. Coelho, L. Pallottino, K. Kondak, and M. J. Kim, "Compliance control of a cable-suspended aerial manipulator using hierarchical control framework," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 7196–7202.

[19] M. Orsag, C. M. Korpela, S. Bogdan, and P. Y. Oh, "Hybrid adaptive control for aerial manipulation," *Journal of intelligent & robotic systems*, vol. 73, pp. 693–707, 2014.

[20] H. Liu, P. Guo, X. Jin, H. Deng, K. Xu, and X. Ding, "An intelligent planning method for the multi-rotor manipulation robot with reinforcement learning," in *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2022, pp. 1028–1033.

[21] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 985–994, 2009.

[22] C. H.-N. Christian Ott, *Grundlagen der Robotik*. ACIN, 2022, vol. 206.

[23] A. STEINBÖCK, *Optimierung*. TU Wien, 2023, vol. 1.

[24] H. Xing, A. Torabi, L. Ding, H. Gao, Z. Deng, and M. Tavakoli, "Enhancement of force exertion capability of a mobile manipulator by kinematic reconfiguration," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5842–5849, 2020.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct - Regeln zur Sicherung guter wissenschaftlicher Praxis (in der aktuellen Fassung des jeweiligen Mitteilungsblattes der TU Wien), insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, im Februar 2024

Sandra Martina Foith