DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Semester Thesis

Extension of the Control Concept for a Mobile Overhead Manipulator to Whole-Body Impedance Control

Leonardo Alessandro Igler

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Semester Thesis

Extension of the Control Concept for a Mobile Overhead Manipulator to Whole-Body Impedance Control

Erweiterung des Regelungskonzepts eines mobilen Überkopfmanipulators auf Ganzkörper-Impedanzregelung

Author:	Leonardo Alessandro Igler
Supervisor:	Prof. Dr.-Ing. Albu-Schäffer Alin Olimpiu
Advisors:	Dr.-Ing. Thomas Hulin
	Dr.-Ing. Martin Görner
	Dr.-Ing. Alexander Dietrich
Submission Date:	01.05.2023

I confirm that this semester thesis is my own work and I have documented all sources and material used.

Munich, 01.05.2023 Leonardo Alessandro Igler

Acknowledgments

This thesis could not have been written, if it was not for the support and guidance I received throughout the creation of my work. I want to take this chance to thank my supervisors, Dr.-Ing. Thomas Hulin, Dr.-Ing. Martin Görner and Dr.-Ing. Alexander Dietrich who accompanied my work for half a year. I would also like to thank Prof. Dr.-Ing. Albu-Schäffer Alin Olimpiu for the possibility to write this thesis.

This work was partially funded by the DLR-internal project "Factory of the Future – Extended (FoF-X)" and by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) as part of Germany's Excellence Strategy - EXC 2050/1 - Project ID 390696704 - Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden.

Abstract

At present, robots constitute a central component of contemporary factories. The application of traditional ground-based systems, however, may lead to congested floors with minimal space left for new robots or human workers. Overhead manipulators, on the other hand, aim to occupy the unutilized ceiling space, in order to manipulate the workspace located below them. The SwarmRail system is an example of such an overhead manipulator. This concept deploys mobile units driving across a passive rail-structure above the ground. Additionally, equipping the mobile units with robotic arms at their bottom side enables this design to provide continuous overhead manipulation while in motion. Although a first demonstrator confirmed the functional capability of said system, the current hardware suffers from complications while traversing rail crossings. Due to uneven rails consecutive rails, said crossing points cause the robot's wheels to collide with the new rail segment it is driving towards. Additionally, the robot experiences an undesired sudden altitude change.

In this thesis, we aim to implement a hierarchical whole-body impedance tracking controller for the robots employed within the SwarmRail system. Our controller combines a kinematically controlled mobile unit with the impedance-based control of a robotic arm through an admittance interface. The focus of this thesis is set on the controller's robustness against the previously mentioned external disturbances. The performance of this controller is validated inside a simulation that incorporates the aforementioned complications. Our findings suggest, that the control strategy presented in this thesis provides a foundation for the development of a controller applicable to the physical demonstrator.

Key Words: Overhead manipulator, impedance control, admittance control, whole-body control, hierarchical control

Contents

1. Introduction

In recent decades, automation has been on the rise in manufacturing and logistics. Hereby, the use of robotic systems has played a significant role in increasing productivity and efficiency in these industrial applications [1, 2]. However, relying solely on ground-based systems limits the possible use of the available workspace. Moreover, applying multiple robots may lead to densely used workshop floors. This in turn implies the impediment of human motion whilst working alongside the respective machines. This is particularly problematic for the cooperative work between humans and robots, which is envisioned for the factories of the future [3]. To address these challenges recent research has proposed to partially uplift robotic systems to the ceiling of workshops hence allowing accessibility from both, the top and bottom.

One representative of this approach is the SwarmRail system [4]. The proposed solution utilizes a rail-network mounted at the ceiling of a workshop, subsequently enabling the mobile robotic systems to operate without any further impediments. Equipping the mobile units with robotic arms, allows for a continuous manipulation of the workspace located below the rails. Moreover, the mobile units are capable of moving without the assistance of active elements inside the rails, allowing the usage of cost-effective and simple grid structures. Additionally, the implementation of multiple robots moving independently from one another inside a single rail-network is thus possible. Consequently, flexible production processes, characterized by a swarm of robots acting simultaneously, are thus enabled.

Although a first demonstrator of the SwarmRail has been constructed at the German Aerospace Center (DLR), the current hardware available serves the sole purpose of demonstrating the mobility of bespoken mobile robots on top of the rails. Therefore, the task of manipulating its environment through the usage of a robotic arm has not been investigated yet. Consequently, the concept lacks a controller for the entire system, which enables it to pursue manipulation tasks on the workspace below.

Motivated by this shortcoming, this semester thesis aims to implement a hierarchical whole-body impedance tracking controller for the SwarmRail system based on the DLR's previous research findings in whole-body control [5–7]. This control strategy allows the robot to complete a multitude of lower-level tasks simultaneously. The controller is then validated inside a physical simulation in which it is subjected to strong and sudden external disturbances. These perturbations are modeled to resemble the issues encountered during the operation of the real-world demonstrator. To the best of our knowledge, this is the first time a whole-body controller is implemented for an omni-mobile overhead manipulator.

The remainder of this thesis is hence divided into seven chapters. Chapter 2 gives

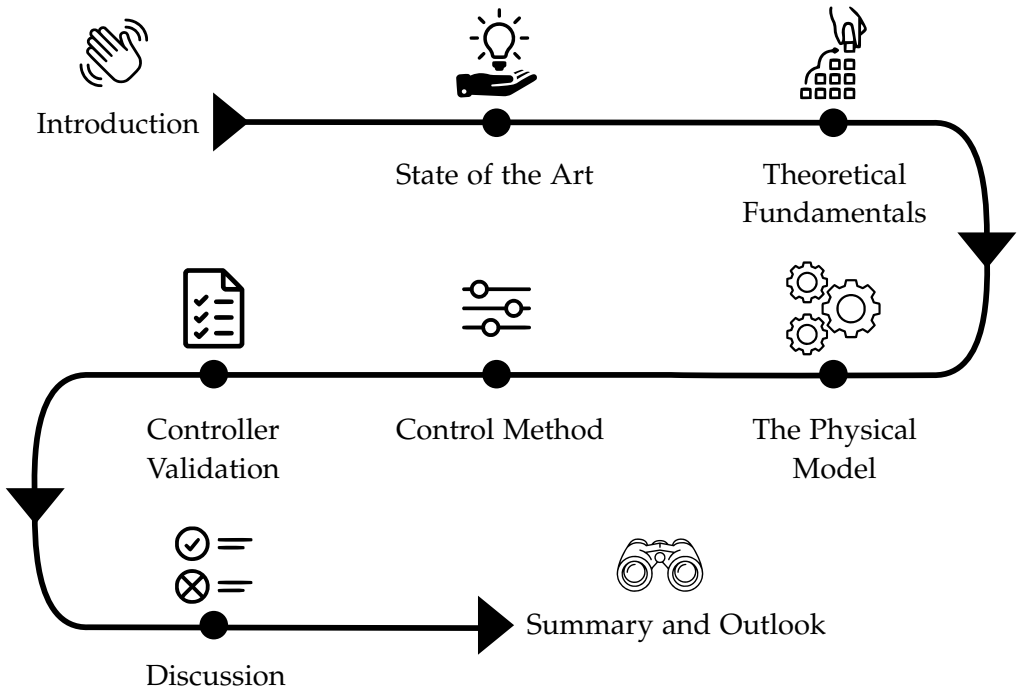


Figure 1.1.: Roadmap for this thesis

a short introduction to hierarchical multi-objective control and provides a selection of existing solutions for overhead manipulators. For the latter, an emphasis is set on the SwarmRail system. Thereafter, the presented controllers and robotic systems are evaluated and compared to each other. In the following Chapter 3 covers the theoretical fundamentals of robot kinematics, dynamics and compliant control theory. Moreover, the fundamental theoretical knowledge needed to comprehend the simulated model and the implemented controller, respectively, is provided.

Beginning with Chapter 4, the work and research conducted during the development of this thesis is covered. Moreover, special attention is given to the introduction of the underlying physical model. Furthermore, an overview of the applied assumptions and simplifications as well as the scenarios, which are investigated to validate the controller's performance, is provided. These basic concepts serve as a foundation for the introduction of the actual controller design, as in Chapter 5. Subsequently, the controller is validated in Chapter 6, providing the results of simulated tracking tasks. A special emphasis is set on the precision of the controller. The simulation results are discussed in Chapter 7. Finally, Chapter 8 gives an overview over future research topics building upon the results of this thesis.

2. State of the Art

In this chapter, an overview of the research progress relevant to this thesis is given. First, the existing solutions for the implementation of hierarchical multi-objective control are briefly presented. Following that, this chapter introduces several concepts for overhead manipulators. Among the overhead manipulators, the SwarmRail system receives a more detailed description, as it represents the robotic system which is being treated in this thesis. Finally, the presented controllers and robotic systems are critically evaluated.

2.1. Hierarchical Multi-Objective Control

For robotic manipulators which possess more Degree(s) of Freedom (DOF) than necessary to pursue a given task, the surplus of DOF can be used in order to incorporate multiple tasks simultaneously [8]. As these tasks might interfere with each other, a task hierarchy is implemented. Therefore, it is meant to guarantee the unhindered execution of the objectives deemed as most important, such as safety-critical tasks [6, 9]. One possibility implies the usage of weight-based "soft" prioritization approaches [10–15]. Given that the priority levels are assigned through the application of weights, these approaches are characterized by the potential of applying flexible prioritization. That is, by a change of the weight parameters, it is thus possible to smoothly modify the task hierarchy order [12, 13].

The usage of null space projectors [16–18] provides an alternative for the multi-objective control. Contrasting the aforementioned method, these strict hierarchies do not assign weights regulating the tasks' influence on in the control actions. Instead, they enforce that lower prioritized control actions do not disrupt tasks of higher priorities. That is, lower prioritized tasks do not cause any control actions unless these actions do not affect the execution of higher prioritized tasks. Therefore, this approach results in a "strict" prioritization [5, 19].

One example of how such a strict hierarchy can be achieved, is given by the Operational Space Formulation (OSF) [20]. This controller utilizes feedback linearization on all hierarchy levels in order to produce linear Equations of Motion (EOM). For these linearized dynamics, OSF has shown to provide exponential stability for trajectory tracking. Moreover, employing "dynamically consistent" null space projectors [21] attains inertial decoupling across all hierarchy levels [6, 19]. Examples for such feedback linearization-based controllers can be found in [22–24].

Alternatively, it can be resorted to the concept of compliance control [25]. This method is to be understood as a variation of the impedance controller developed by Hogan

[26–28], with the difference that the system's natural nonlinear inertia is preserved. Dietrich et al. [6] proposed such a controller for a strict prioritized task hierarchy. For the case of no external torques acting on the system, they were able to prove asymptotic stability of the equilibrium. The latter is defined as the state in which the operational space tracking errors for all hierarchy levels are equal to zero. By including a term compensating for external forces/torques, the asymptotic stability is also valid at the presence of constant external forces. This controller is covered in extensive detail in Section 5.2.

2.2. Overhead Manipulators

The concept of utilizing the ceiling in order to incorporate robotic manipulators acting on a workspace below, has already been subject of previously conducted research. One example of such an implementation, can be found in gantry systems, such as the ceiling-mounted home robot by the Toyota Research Institute [29]. Actuated by two motors, the robot arm moves along two linear axes to reach a desired position in the room and interacts with the workspace beneath it. This system aims to aid humans on everyday tasks, such as cleaning or transporting objects. Being mounted above the ground, the robotic system is supposed to avoid navigating through floor clutter and cramped spaces. Moreover, when not in use, the overhead design allows the robot to tuck itself up, not disturbing the homeowners.

Another example of a gantry-based robotic overhead manipulator is the Multifunctional Cell (MFZ), developed by the DLR's Center of Lightweight Production Technology in cooperation with KUKA Systems GmbH, Augsburg [30]. The proposed solution implements five robots moving on linear tracks mounted at the ceiling of a workshop. The robots moving across the ceiling are set up in such a way, that their respective individual workspaces overlap thus enabling cooperative work between the system-bound robots. The available workspace is separated into different cells with each cell being assigned to a particular configuration of robots.

As an alternative to gantry systems, the autostore system [31] can be named. A swarm of robots, each equipped with eight wheels, moves across a rectangular grid structure, located at the top of stacked store bins. The robots are capable of autonomously performing translations in two orthogonal directions, only using four of their eight wheels for each direction. Thereby, at the crossing points, the robot elevates four wheels which it does not use and descends the four wheels it uses for the designated translation direction. After moving to their intended position at a desired column, they interrupt their motion to reach downwards to store or retrieve a new bin to their column.

2.3. The SwarmRail System

The SwarmRail system, proposed by Görner et al. [4], is another example of a robotic overhead manipulator. Hereby, a swarm of mobile robots is supposed to be placed on a

rail grid, mounted at the ceiling of the workshop. The robots are capable of traversing the grid tracks and change their direction of motion at intersections, without the need for active elements on the rails, such as track switches. The SwarmRail system complements or replaces ground-based systems and yields the following benefits:

- Continuous overhead manipulation, even whilst traversing rail crossings.
- Flexible application possibilities due to the ability to deploy multiple mobile units, acting independently inside a single rail-network.

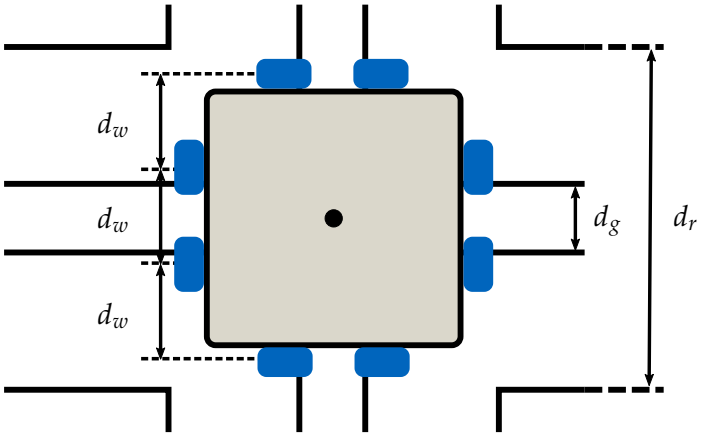


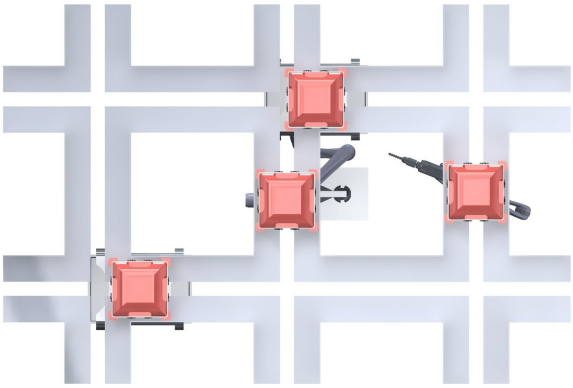
Figure 2.1.: Top view of the SwarmRail system located at a crossing point. Figure adapted based on Görner et al.[4].

The following subsections introduce the rail-network and the mobile unit, respectively.

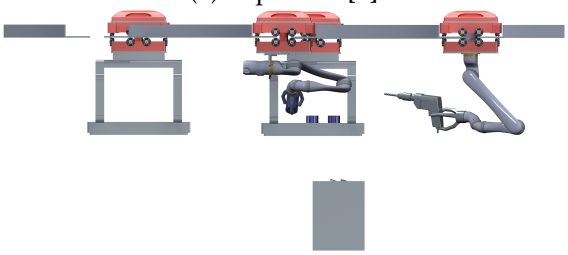
2.3.1. Rail-Network

The rail-network developed as part of the SwarmRail system, is characterized by a collection of straight rail sections, further denoted as "straight segments", which merge into each other in the form of intersecting gaps at the crossing points. The mentioned straight segments are constructed as two L-profiles facing each other. These profiles are separated by a through-going central gap of length d_g, which guides the motion of the mobile unit. Furthermore, the vertical elevation of the L-profiles provides a physical border for the robots. Herewith, the robots' range of motion inside the straight segment is limited to translation in two directions. The crossing points provide the system with additional mobility possibilities, enabling the robots to change their direction of translation or perform rotations.

The SwarmRail system allows for a multitude of different potential grid shapes, such as hexagonal, rectangular or quadratic [32], as long as two conditions are met. First, the rail width d_r must be large enough in order to allow the robot's movement through the rails. Expressed differently: the distance d_r between the vertical elevations of facing



(a) Top View [4]



(b) Side View [4].

Figure 2.2.: Visualization of a potential rectangular grid system with multiple SwarmRail units operating and cooperating together, using robotic arms and transportation boxes fixated at the bottom of the respective mobile unit. Figures adopted from Görner et al. [4].

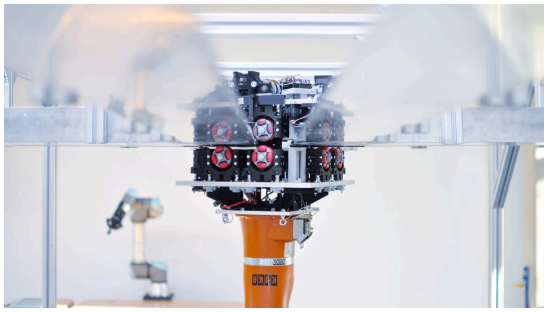
L-profiles needs to be wider than the edge sizes of the robot. Secondly, the gap size d_g needs to be smaller than the distance between two adjacent wheels d_w (refer to Figure 2.1). This serves the purpose, to ensure the contact of at least two driving wheels with the rail, allowing the robot to remain mobile at all times.

However, this thesis only considers the simplest structure, which also corresponds to the network demonstrator built at the DLR: a rectangular grid structure (refer to Figure 2.2).

2.3.2. Mobile Unit

The mobile unit consists of two elements connected through the ongoing gap, which extends throughout the entire rail-network (refer to Figure 2.3b). All in all, the top and the counterpart at the bottom of the rail, have the same kinematic setup: both systems possess eight omni-wheels. These wheels are actuated by four brushed Direct-Current (DC) gearmotors for the top part. The counterpart on the bottom remains completely passive. Hereby, each motor is responsible for the wheel pair of one edge. This configuration allows for omnidirectional mobility, while keeping the number of actuators to a minimum. Figure 2.4 demonstrates the unit's movement based on the

(a) The first demonstrator of the SwarmRail system, with a LWR attached at the bottom of it [36].



(b) Closeup of SwarmRails mobile unit, consisting of two parts located at opposite sides of the rail [36].

Figure 2.3.: Different perspectives on the robotic demonstrator built and located at DLR

rotation direction of each omni-wheel pair.

Despite its lack of actuating elements, the lower part is essential for the avoidance of tilting while the robot is exposed to high external torques. Additionally, the bottom part provides the possibility of incorporating an attachment at the mobile unit, allowing the robot to perform tasks at the workspace below it. Depending on the task to be performed, various attachments are imaginable to complement the robot, such as boxes and robot arms (refer to Figure 2.2). However, for the remainder of this thesis, only the case of the attachment being a robotic arm is considered.

While the mobile unit is responsible for the robot's movement inside the rail-network, the robotic arm attached at the bottom of the mobile unit allows the SwarmRail robot to manipulate its environment and to perform operations such as pick and place tasks. It is important to note, that the current SwarmRail demonstrator built at the DLR, portrayed in Figure 2.3a, already consists of a mobile unit which allows a robotic arm to be mounted. However, the arm is not functional yet and the entire construction has not yet been tested on real tasks. The available hardware solely serves the purpose, of demonstrating the mobility of said subsystem in a rectangular rail-network. The robotic arm which is expected to be mounted on the mobile units, is a variation of the Lightweight Robot (LWR) [33–35] developed by DLR. This subsection however focuses on the novel mobile unit developed by Görner et al. [4].

As already covered in Subsection 2.3.1, the special shape of the rail profiles provides a physical barrier for the robot, limiting its motion to a translation along the rail. However, beyond these physical borders, the mobile unit is able to center itself with regard to the gap extending throughout the rails. Hence, the physical contact between the rail's vertical elevation and the unit is avoided. Therefore, the robot is equipped with four sensors based on Infrared (IR), each sensor being located at the center of each of the four sides of the robot. These sensors are further denoted as the "gap detection sensors".

The gap detection sensors emit IR radiation towards the surface located below them, i.e. on the grid section the mobile unit is currently located at. The lights is reflected back

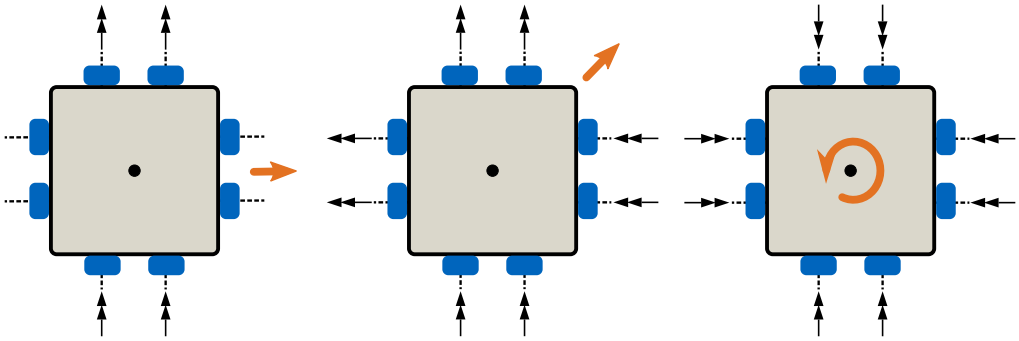


Figure 2.4.: Based on the chosen rotation direction of the wheels, the mobile unit is capable of performing translations in arbitrary directions or rotating about the vertical axis. The double headed vectors express the rotation of the omni-wheels, the orange vectors represent the overall motion of the mobile unit.

to the sensors and allows the determination of the mobile unit's position relative to the gap. For the case of the mobile unit being perfectly centered, the emitted light passes through the gap. Consequently, only a small portion of the light is reflected back to the sensors. Should the robot not be centered, the IR radiation becomes partially directed at the metal rails beneath the sensor. Therefore, the metallic surface reflects a larger portion of the emitted light back to the sensors. Based on the direction from which the reflected light reaches the sensors, the readings provide information regarding the direction to which the position should be corrected, namely in the direction opposing the source of the reflected light. Therefore, a translational motion does not solely make use of the driving omni-wheels, whose rotation axes are orthogonal to the direction of the movement. Additionally, the wheels whose rotation axes are parallel to the translation are also actuated according to their respective gap detection sensors. By that, the robot is centered about the gap. The centering inside the rails is achieved via a Proportional–Integral–Derivative (PID) controller [37], implemented by Grimmel [38].

The centering along the straight segments only made use of two sensors, both at opposing sides of the robot but located above one single ongoing gap. However, when the robot approaches a crossing point, the problem increases in its complexity. At the crossing points of a rectangular grid structure, four different straight segments with four different gaps meet (refer to Figure 2.2a). Therefore, the centering now takes place based on all four gap-detecting IR-sensors. The transition from these two centering-methods is initiated through the classification of the robot's current position as either inside a straight segment or a crossing point. This is performed using eight additional IR-sensors. These sensors attached at the four corners of the robot, detect whether the robot's edges are surrounded by rails or not. These sensors shall be named "edge detection sensors". For a robot inside a straight segment, these edge detection sensors detect the reflection caused by the two opposing vertical elevations of the L-profiles making up the segment. Should none of the sensors detect reflection of IR-light from their respective sources, the

robot is assumed to be inside a crossing.

The aforementioned interpretation of the own position is not only necessary for the centering, it also allows the robot to detect which motion possibilities are limited. Inside the straight sections of the network, the robot is only able to perform translation in one of two directions. However, when located inside a crossing, the robot can drive in one of four directions or rotate by setting all eight wheels to rotate in the same direction (clock- or counterclockwise).

Additionally, the mobile unit possesses an Inertial Measurement Unit (IMU). A summary of essential technical data can be found in Table 2.1.

Table 2.1.: Summarized technical properties of the mobile unit. Table is partially adopted from Görner et al. [4].

System specifications	
Size in mm	362×362×263 (length × width × height)
Mass in kg	17.5
Motors	4 × brushed DC gearmotor
Maximal Speed in m/s	0.5
Wheels	8 × driven omni-wheels 8 × passive omni-wheels on lower part
Sensors	4 × IR gap detection 8 × IR edge detection 1 × IMU

2.4. Discussion of the State of the Art

The implementation of weight-based soft prioritization introduces couplings between the tasks of different hierarchy levels. Although the impact of each task can be adjusted, superposing the tasks leads them to compete, to some extent, against one another [6]. The control approaches based on feedback linearization, on the other hand, allow for a clear separation between tasks. Thereby, it enables the best achievable performance on the tasks of highest priority. Moreover, exponential stability for trajectory tracking can be proven. However, if a desired impedance is to be implemented in this approach, a precise knowledge of external forces/torques and dynamical parameters underlying the analyzed system is required [6, 19]. Apart from the difficulty of obtaining reliable and accurate sensor readings of external forces/torques, these approaches are sensitive against modeling uncertainties and parameter variations [19].

On the other hand, compliance control as implemented by Dietrich et al. [6], mainly benefits from the fact of neither requiring feedback of external forces nor a precise model of the inertia matrix. This in turn has proven to be advantageous in terms of stability and control performance [25, 39]. However, the controller developed in [6] only

provides local asymptotic instead of exponential stability. Furthermore, the application of dynamic external torques is shown to introduce cross-couplings between tasks of higher with tasks of lower priorities, respectively. The latter is addressed by Wu et al. [19]. The details concerning [6, 19] are provided and thoroughly explained in Chapter 5.

Regarding the overhead manipulators, it can be said that gantry-based systems, such as the ceiling-mounted home robot by the Toyota Research Institute [29] are limited in their applicability. The linear positioning actuators do not allow the implementation of numerous robots moving independently across the ceiling. Although propositions, such as the MFZ, are able to combine multiple gantry-based manipulators, the respective robots are limited in their motion capabilities. These systems enable translations along a linear track only. Consequently, the robots are unable to cover the ceiling space as a whole by themselves. Instead, they rely on the partition of the available workspace in cells. By doing so each cell is then individually covered by a particular configuration of robots. Be that as it may, the construction of such actuated ceiling constructions remains to be costly and complex for both gantry-based systems.

The autostore system [31] on the other hand, does not make use of any active elements inside the rail-network it moves across. Therefore, the system is capable of operating a swarm of robots simultaneously. However, these robots are not capable of moving while lifting goods or manipulating their environment.

Similarly to autostore, the robots of the SwarmRail system [4] move autonomously without the need of actuators integrated inside the rail-network. Therefore, this system is also capable of operating a swarm of independent mobile units at the same time. Furthermore, the mobile units provide continuous overhead manipulation at all times. However, while the autostore system is already in use for industrial applications, the SwarmRail system is currently still in the beginning of its developing phase. Consequently, the system still possesses flaws, for which solutions are still to be discovered.

For instance, the currently built SwarmRail demonstrator does not provide consistent grip between its wheels at the rails, which is particularly apparent while the mobile units enter crossing points. Consequently, the robot has to reduce its speed while entering crossing points from the maximal speed of $0.5\frac{\mathrm{m}}{\mathrm{s}}$ to $0.3\frac{\mathrm{m}}{\mathrm{s}}$ [4, 38]. The source for this can be mainly attributed to two factors. Firstly, rails are too smooth for the wheels to have enough traction to reliably drive the robot forwards. Our experience has shown, that regularly roughening the rails is able to counteract this problem. Secondly, the undesired behavior can also be caused by manufacturing or construction inaccuracies of the rails. These conditions cause the rail segments to differ from each other regarding their altitude. Consequently, for different wheels having contact to rails at different altitudes, the contact forces are no longer equally distributed between them. Should the driving wheels have particularly small contact forces, the robot is unable to generate the necessary force to move. Moreover, differing driving forces for the wheels are also able to cause undesired rotations of the mobile unit, despite applying the same motor torques on both sides. This has as a consequence, that constant intervention by the PID controller is necessary to prevent the undesired rotation and center the robot.

An additional implication of the differing rail altitudes, becomes visible while the robot is traversing a crossing point. During our experiments with the system, we were able to observe that the robot experiences collisions while crossing the intersection. These collisions occur when the wheels hit the edge of the rail gap orthogonal to the direction of motion. The exposition to these problems has motivated us to further reduce the mobile unit's speed, in order to guarantee its stability inside crossing points. For our investigations, the mobile unit has shown to be reliable at speeds ranging from $0.1\frac{\mathrm{m}}{\mathrm{s}}$ up to $0.25\frac{\mathrm{m}}{\mathrm{s}}$.

Current experiments with the SwarmRail have also shown a new problem, regarding its centering capabilities. Being reliant on the reflective properties of the rails, the gap detection sensors are sensitive to lighting conditions. Bright lights or impurities on the rails affect the perceived light reflection and disturb the system, preventing it from centering itself. This problem is particularly troublesome for the crossing points, in which the centering occasionally fails.

Lastly it should also be noted, that despite the SwarmRail being envisioned as an overhead manipulator, the only controller currently developed is the PID controller responsible for the centering of the mobile unit inside the rails. The realization of overhead manipulation has been left unexplored until now.

Summarizing it can be said, that in a limited set of applications, overhead manipulators have shown to be possible alternatives to ground-based systems. Making use of the commonly unused space above, overhead manipulators are able to exploit the workspace more efficiently, enabling the access to the workspace from two directions. Moreover, transferring robots to the ceiling allows for less cluttered floors and consequently more space for humans to move unhindered. Therefore, overhead manipulators also possess the potential to be particularly beneficial for the cooperative work with humans.

3. Theoretical Fundamentals

This chapter aims to explore the fundamental concepts which underlie the hierarchical whole-body impedance control, the implementation of which is the subject of this thesis. By establishing the very basic definitions and mathematical formulations of both, robot kinematics and dynamics, the fundamental knowledge behind robot motion is set at the beginning of this chapter. Building upon this, the reader is introduced to the basic principles behind compliant control. In doing so, the concepts of impedance- and admittance-based control methods are presented. This chapter is finalized with an introduction to the concept of tasks hierarchies for redundant systems, which is an essential component for the control method implemented in this thesis.

3.1. Rigid Body Kinematics

The field of kinematics concerns itself with the description and analysis of time-dependent properties of motion. However, the underlying causes find no further analysis. This section is mainly inspired by the works of Craig [40], Murray et al. [41] and Siciliano et al. [8]. It aims to introduce the concepts which are essential for the comprehension of the controller, as implemented in this thesis. For a more detailed study, the reader is thus advised to reference the aforementioned authors' extensive work.

Starting off by its definition, a rigid body is a system in which the distances between any two of its particles remains constant. These distances are thus independent of both the motion performed and the forces being exerted on the body. Therefore, the kinematics of a rigid body can be described entirely by its position and orientation as well as their respective time derivatives [8, 41].

In order to examine a motion, a so-called "reference frame" is established. This reference frame, for the Cartesian space, is defined by a point of origin and a set of w straight lines, the axes, which intersect at the point of origin [42]. Moreover, the axes are selected to be orthogonal to each other. Through these axes, the position of a point is defined by the distance between the origin and the point's projections onto each of the w axes. Therefore, a combination of w real numbers, expressing the point's distances in the direction of the axes, is able to fully define its position. For the description of bodies inside a system with w mutually orthogonal axes, customarily, a set of w orthonormal vectors is used. The term orthonormal states that these vectors are not only orthogonal to each other (refer to Section A.1), but also have a magnitude of 1. Thus, for the purpose of describing movements inside the three-dimensional space, a set of three axes is able

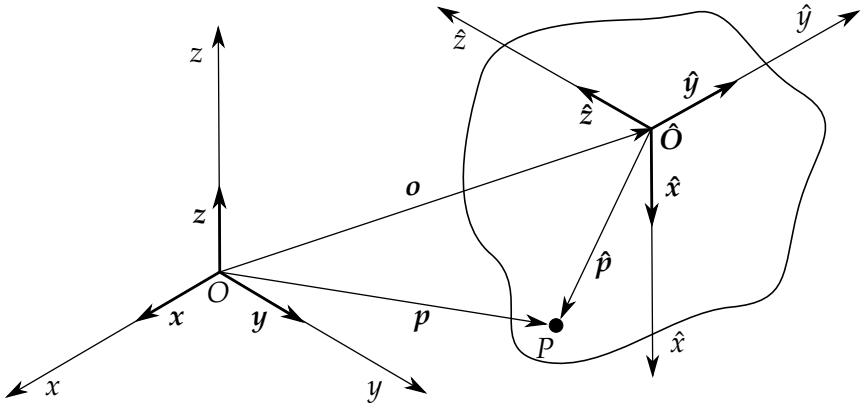


Figure 3.1.: The coordinate frame O-xyz can be transferred to the second coordinate frame $\hat{O}$-$\hat{x}\hat{y}\hat{z}$ by a translation of the origin by $\boldsymbol{o}$ and rotation of the respective axes. Performing said transfer also provides knowledge on how to express points in different frames. This figure is based on Siciliano et al. [8].

to fully describe the behavior of physical bodies.

For the three-dimensional example of Figure 3.1, the frame O-xyz is defined by the origin O and the axes x, y and z, respectively. These axes, in turn, are represented by the unit vectors $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^3$. A point in space, such as the illustrated point $\hat{O}$ (refer to Figure 3.1), can be described with respect to the frame O-xyz, by the vector from the frame's point of origin O to $\hat{O}$. This vector shall be denoted as $\boldsymbol{o}$. The description of said vector is achieved by making use of the unit vectors. Hence, the projections onto the three axes are multiplied with the corresponding representative unit vector and summed up, taking the form

$$\boldsymbol{o} = o_x\boldsymbol{x} + o_y\boldsymbol{y} + o_z\boldsymbol{z} \quad . \tag{3.1}$$

Typically, this combination of projection values and axes is expressed inside a single vector. For the case of $\boldsymbol{o}$, this would yield

$$\boldsymbol{o} = \begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix} \in \mathbb{R}^3 \quad . \tag{3.2}$$

Expanding these concepts from a single point to a body is straightforward for the case of rigid bodies. As all points making up the body preserve their respective distances to each other, under the knowledge of the body's spatial dimensions, being able to describe one point of the body gives access to the positions of all the other points as well. Therefore, the description occurs through the definition of a new frame with three orthogonal axes represented by three orthonormal vectors, respectively. This frame, however, is fixed on a point of a body and is hence referred to as the "body-frame". In the case of Figure 3.1 this very frame is given by $\hat{O}$-$\hat{x}\hat{y}\hat{z}$ [8]. The position of any particle inside the described body can be derived from the particle's position with respect to

the body-frame. If the position of the body-frame's point of origin with respect to a reference frame is also known, the motion of all points making up the body can, hence, be fully described inside said reference frame [41].

Furthermore, by introducing a second coordinate frame it is possible to expand the description of the body beyond its position: by comparing the respective individual axes of both frames, the orientation of the body-frame can be discovered. Consequently, the body on which the body-frame is attached, can also be defined solely by its position and orientation. The combination of both, position and orientation, is from now on denoted as the "pose" of the body. Similarly to the relation described in (3.1), determining the orientation also occurs via a projection onto the axes of the reference coordinate frame. However, for the orientation, three projections of the body-frame's axes onto the respective three axes of the reference frame are examined. These projections are defined as

$$\begin{aligned} \hat{\boldsymbol{x}} &= \hat{x}_x \boldsymbol{x} + \hat{x}_y \boldsymbol{y} + \hat{x}_z \boldsymbol{z} \\ \hat{\boldsymbol{y}} &= \hat{y}_x \boldsymbol{x} + \hat{y}_y \boldsymbol{y} + \hat{y}_z \boldsymbol{z} \\ \hat{\boldsymbol{z}} &= \hat{z}_x \boldsymbol{x} + \hat{z}_y \boldsymbol{y} + \hat{z}_z \boldsymbol{z} \quad , \end{aligned} \tag{3.3}$$

and represent the orientation of the body-frame relative to the reference frame. The aforementioned results can be expressed through the usage of the rotation matrix

$$\boldsymbol{R} = \begin{bmatrix} \hat{\boldsymbol{x}} & \hat{\boldsymbol{y}} & \hat{\boldsymbol{z}} \end{bmatrix} = \begin{bmatrix} \hat{x}_x & \hat{y}_x & \hat{z}_x \\ \hat{x}_y & \hat{y}_y & \hat{z}_y \\ \hat{x}_z & \hat{y}_z & \hat{z}_z \end{bmatrix} \in \mathbb{R}^{3\times 3} \quad , \tag{3.4}$$

which stacks these projections horizontally inside a single matrix.

An alternative variant of defining orientations, is achieved through the usage of Euler angles. On this matter, an arbitrary rotation required to align two frames is decomposed into a set of three elementary rotations about three axes. It must however be guaranteed that two consecutive rotations are not performed about parallel axes. By making use of this representation, a minimal description of the orientation change is achieved. The number of parameters needed to describe an orientation in a three-dimensional space is consequently reduced from nine to three [8]. Moreover, Day [43] describes a method of extracting the rotation angles with respect to the three different axes of a stationary reference frame.

Combining the translation vector $\boldsymbol{o}$, expressed in O-xyz, with the rotation matrix $\boldsymbol{R}$, allows the description of the transition between the corresponding coordinate frames. Furthermore, any point whose position relative to one frame is known can, as a result, be described with respect to the other frame as well. For instance, let's take an arbitrary point P, described as $\hat{\boldsymbol{p}}$ by frame $\hat{O}$-$\hat{x}\hat{y}\hat{z}$. This very point can be expressed with respect to the frame O-xyz, resulting in the expression

$$\boldsymbol{p} = \boldsymbol{o} + \boldsymbol{R}\hat{\boldsymbol{p}} \quad . \tag{3.5}$$

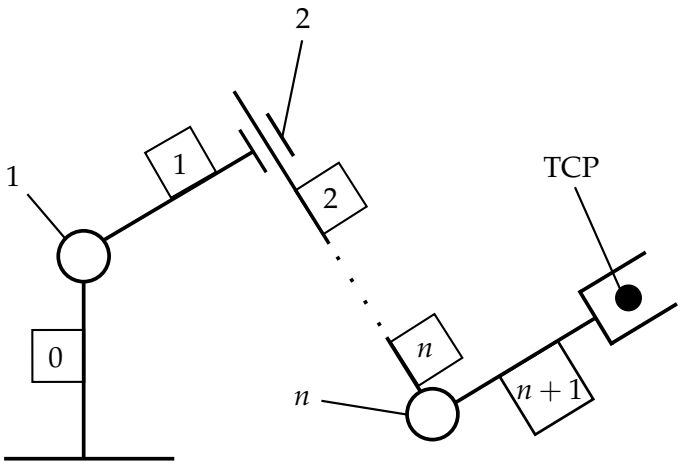


Figure 3.2.: Illustration of an open kinematic chain.

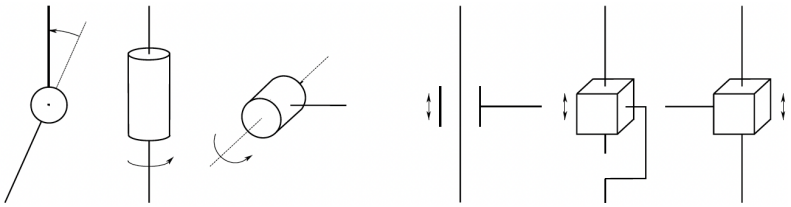


Figure 3.3.: Illustration demonstrating revolute and prismatic joints as well as the motion they enable for connected links [8].

This very transition between descriptions, with respect to different coordinate frames, is commonly used in the field of robotics. As a matter of fact, a robotic manipulator is typically comprised of a sequence of interconnected bodies or links which are held together by joints (please refer to Figure 3.2). These joints do not solely serve as physical connections between the bodies, but also as pivotal points that enable the robot to perform movements.

Most manipulators make use of two categories of one-dimensional joints: revolute and prismatic joints [41]. While revolute joints allow relative rotations between connected links, a translational sliding motion is achieved through the usage of prismatic joints (please refer to Figure 3.3). Although joints with multiple DOF exist, any joint with k $\{k \in \mathbb{N} | 1 \leq i \leq 6\}$ DOF can be modeled through a combination of k revolute or prismatic joints [40]. Therefore, this thesis only considers these two types of joints. Moreover, since it corresponds to the structure of the robotic system treated in this work, this thesis is restricted to one particular type of robot construction. In this setting, there is only one sequence of links connecting the first link to the last without the presence of loops. This structure is referred to as an "open kinematic chain" [8].

Consider the robotic manipulator illustrated in Figure 3.2. Here, the robot consists of $n + 1$ links, connected by n one-dimensional joints. Each joint connects two consecutive

links, therefore defining the kinematic relationship between both, consecutively. Starting from link 0, which is conventionally immobile and fixed to the ground, the index of the first moving link is 1, the second link has the index 2 and so on, until the free end at link $n+1$ is reached [40]. Iterating recursively between all links of the operator enables the overall description of the manipulator's kinematics [8]. Therefore, defining a coordinate frame attached to each link from 0 to $n+1$ allows to determine the pose of each link with respect to an arbitrary link's frame, as proposed by (3.4) and (3.5). Above all, a description of the nth frame's orientation and position with respect to frame 0 is, thus, obtained [8].

This represents a particularly important scenario since the environment and tasks to be performed in it are commonly described in the frame fixed to the environment, namely frame 0. However, the last frame is of special relevance, since it describes the very end of a robotic manipulator. The latter usually incorporates tools such as grippers in order to interact with its surroundings [40]. The tool is represented by a point, which is referred to as the "Tool Center Point (TCP)".

Given that the spatial dimensions of the robot are known, the robot links' pose can be fully described by the angles and translations of the n respective joints. These angles and translations are summarized inside a single vector

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \in \mathbb{R}^n \quad , \tag{3.6}$$

which is referred to as the "joint vector" $\boldsymbol{q}$ [40].

Depending on which type of joint is being represented each element q_i, with $\{i \in \mathbb{N} | 1 \leq i \leq n\}$, enables either a relative rotation or translation between two consecutive links. The values assumed by the joints are henceforth referred to as "joint values". The space of all possible values for these joint variables is consequently referred to as the "joint space" [41].

Although the values for the joints are sufficient for a full description of the robot's motion, they are usually not used in order to define a task to be fulfilled by the robot. Instead, the task is formulated inside the task-oriented or operational space [40]. This minimal mathematical description of the desired behavior, sets the focus solely on the very essential elements guaranteeing the success of bespoken task. For instance, the objective treated in this section, namely, reaching a point in space or following a defined trajectory would commonly be expressed inside the Cartesian space. Finding the mapping from the joint space to the operation space is task of the forward kinematics. Analogously, the mapping from the operation space to the joint space is treated by the field of inverse kinematics. In this setting, however, no description of how the task shall be completed are set [8, 44, 45].

For a variable $\boldsymbol{x}$ specified inside the operational space, such as the position of the TCP, the relationship between the operational and joint space can be expressed as

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q}) \in \mathbb{R}^m \tag{3.7}$$

$$\dot{\boldsymbol{x}} = \frac{\delta \boldsymbol{f}(\boldsymbol{q})}{\delta \boldsymbol{q}} \dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q}) \dot{\boldsymbol{q}} \tag{3.8}$$

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q}) \dot{\boldsymbol{q}} \quad . \tag{3.9}$$

Let $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$ denote the Jacobian matrix (please refer to Section A.2). This matrix is of great relevance throughout the remainder of this thesis.

3.2. Rigid Body Dynamics

The previous section outlined a simplified workflow for a robotic manipulator. Beginning with the description of a task inside the operational space it is thus required to compute the respective joint values needed to fulfill a given task. It is hence implicitly assumed that once the necessary joint configuration is determined, the robot's joints would faithfully be set according to the desired configuration [41]. This explanation, however, omits how the motion of joints is caused in the first place.

The process of configuring joints in a desired manner is achieved through the application of wrenches (torques and forces) [41] on the joints, via the usage of electric, hydraulic or pneumatic actuators. What is more, the field of dynamics has its objective in describing how a robots motion is affected by the application of wrenches [41].

The dynamic equations of a rigid robot with n DOF, are expressed by the EOM. Inside the joint space they are given by

$$\boldsymbol{M}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}^{\text{ext}} + \boldsymbol{\tau} \quad , \tag{3.10}$$

with $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ being the first and second derivatives with respect to time, respectively. The matrix $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is hence the positive definite and symmetric inertia matrix. All of the aforementioned properties can thus, according to Petersen et al. [46], be expressed mathematically as

$$\boldsymbol{x}^T \boldsymbol{M}(\boldsymbol{q}) \boldsymbol{x} > 0, \quad \forall \boldsymbol{x} \neq \boldsymbol{0} \tag{3.11}$$

and

$$\boldsymbol{M}(\boldsymbol{q}) = \boldsymbol{M}(\boldsymbol{q})^T \quad \text{, respectively.} \tag{3.12}$$

The vector $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ contains all the gravitational wrenches acting on the joints, and can be calculated via the gravity potential $V_g(\boldsymbol{q})$ as: $\boldsymbol{g}(\boldsymbol{q}) = \left(\delta V_g(\boldsymbol{q}) / \delta \boldsymbol{q}\right)^T$.

External wrenches acting on the joints are expressed as $\boldsymbol{\tau}^{\text{ext}} \in \mathbb{R}^n$. The actuator's outputs, and therefore the capabilities of enforcing a desired behavior of the robot, are contained inside $\boldsymbol{\tau} \in \mathbb{R}^n$. The vector $\boldsymbol{\tau}$ contains both, forces and torques acting on the joints. While the former affect revolute joints, the latter are applied on their prismatic

counterparts. Nevertheless, the vector $\boldsymbol{\tau}$ is referred to as the "joint torque vector" for the remainder of this thesis.

Coriolis and centrifugal effects are contained in the product $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$. The definition of the Coriolis matrix $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is generally not unique. For the scope of this thesis, the matrix is chosen according to Murray et al. [41]. Accordingly to this definition, the element of the ith row and jth column of the Coriolis matrix is defined as

$$\boldsymbol{C}_{i,j}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2} \sum_{k=1}^{n} \left(\frac{\delta M_{i,j}}{\delta q_k} + \frac{\delta M_{i,k}}{\delta q_j} - \frac{\delta M_{k,j}}{\delta q_i} \right) \dot{q}_k \quad . \tag{3.13}$$

This particular choice has two equivalent properties: the skew symmetry of $\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ (please refer to Section A.3) and

$$\dot{\boldsymbol{M}} = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T \quad . \tag{3.14}$$

The interested reader is hence advised to refer to Section B.1 for further insights regarding the proof of equivalence for both properties. What is more, the consequent mathematical proof which confirms these properties can be found in Section B.2.

The EOM are formulated inside the joint space, therefore considering the application of joint torques on the robot. The transfer from joint torques to wrenches in the operational space can be achieved through the Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ with

$$\boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{F} \quad , \tag{3.15}$$

where $\boldsymbol{F} \in \mathbb{R}^m$ represents the wrenches inside the operational space.

3.3. Compliant Control

The preceding section introduced the concept of using motors to provide torques, whilst consequently setting the robot in motion. However, it did not provide an answer to the question of how to choose the joint torques in order to fulfill a desired task. This problem is addressed by the implementation of a controller for the robotic system [41] which is the subject of this section.

More precisely, this section sets its focus on control methods which realize compliant behavior of the robotic manipulator. These methods are of special relevance in applications in which the robot interacts with unknown environments or in proximity to humans. Throughout this section, impedance- and admittance-based controllers are introduced. Apart from solely reaching desired positions and joint configurations these approaches also ensure a desired response to disturbances caused by external forces [44]. It shall be mentioned, that the focus of this section lies in giving a broad overview of the underlying idea inspiring both methods: the actual implementation and combination of both methods is covered in Chapter 5.

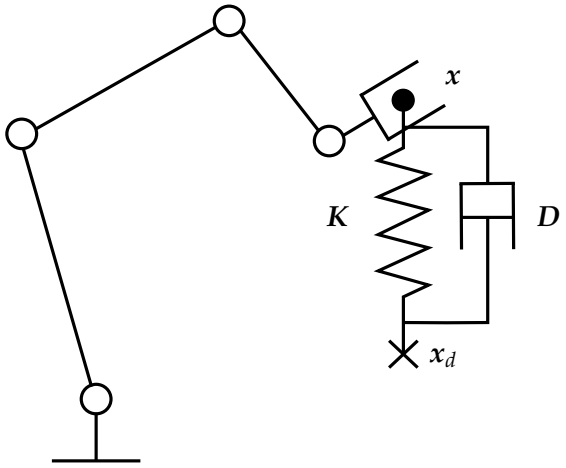


Figure 3.4.: For the example of TCP position control, the tip of the robot is set to behave as a Cartesian spring-damper system, with x_d serving as the force-free point of equilibrium. Figure adapted from Haddadin [48].

3.3.1. Impedance Control

Impedance control, originally developed by Hogan [26–28], has as its objective in realizing a dynamical relationship between the manipulator's motion variables (displacements, velocities or accelerations) and external wrenches [27, 44]. In this thesis, motivated by the benefits presented in Section 2.4, the focus is set on special variation of impedance control, namely compliance control [25]. That is, a variation of the classical impedance controller with natural inertia preservation.

The basic concept of impedance-based control can be explained by the application of a virtual spring-damper directing a task variable to its equilibrium [47]. For the case of the impedance-based Cartesian position control of the TCP, the approach can be comprehended via the visualization in Figure 3.4.

A virtual spring-damper system is implemented between the TCP, located at a position $\boldsymbol{x} \in \mathbb{R}^m$ and a desired position $\boldsymbol{x}_d \in \mathbb{R}^m$. The equilibrium of the spring, i.e. the point at which the spring does not exert any force on the TCP, is reached at the desired position $\boldsymbol{x}_d$. Therefore, the magnitude of the Cartesian error $\tilde{\boldsymbol{x}}$ influences the force acting on the TCP directly. The error is hence defined as

$$\tilde{\boldsymbol{x}} = \boldsymbol{x}(\boldsymbol{q}) - \boldsymbol{x}_d \in \mathbb{R}^m \quad . \tag{3.16}$$

The potential of the virtual spring is given by

$$V_{k,\text{imp}}(\tilde{\boldsymbol{x}}) = \frac{1}{2}\tilde{\boldsymbol{x}}^T \boldsymbol{K} \tilde{\boldsymbol{x}} \quad . \tag{3.17}$$

Consequently, the force exerted by the virtual spring on the TCP can be expressed as

$$\boldsymbol{F}_{k,\text{imp}} = -\left(\frac{\delta V_{k,\text{imp}}(\tilde{\boldsymbol{x}})}{\delta \boldsymbol{x}}\right)^T \quad . \tag{3.18}$$

Put differently, once the TCP's position deviates from the desired location, it applies a force to its attachment. The larger the error, the stronger the force.

On the other hand, the damping element of the impedance results in the force

$$\boldsymbol{F}_{d,\text{imp}} = -\boldsymbol{D}\dot{\boldsymbol{x}} \quad . \tag{3.19}$$

The corresponding joint torque vector caused by the virtual spring-damper can be obtained through the application of (3.15), which yields in

$$\boldsymbol{\tau}_{\text{imp}} = \boldsymbol{J}(\boldsymbol{q})^T \left(\boldsymbol{F}_{k,\text{imp}} + \boldsymbol{F}_{d,\text{imp}}\right) \quad . \tag{3.20}$$

Analogously, using the same principle of applying a virtual spring-damper on the robot, a wide variation of objectives beyond position control can be achieved, such as reaching a desired orientation for the robot. Alternatively, the spring-damper can also be applied in the joint space, specifying desired joint configurations to be reached.

3.3.2. Admittance Control

While the impedance controller measured motion variables and derived corresponding torques acting on the robot, the admittance controller operates contrarily: based on the measurement of external wrenches achieved by force-torque sensors, the controller generates motion variables [8, 44]. One way of achieving this in a robotic system, is by defining a dynamic model which reflects the desired dynamics, as seen in (3.21). This very technique is also applied in the controller implemented in this thesis. The desired dynamic model is given by

$$\boldsymbol{M}_{\text{adm}}\ddot{\boldsymbol{x}}_{\text{des}} + \boldsymbol{D}_{\text{adm}}\dot{\boldsymbol{x}}_{\text{des}} = \boldsymbol{F}^{\text{ext}} \quad , \tag{3.21}$$

with $\boldsymbol{x}_{\text{des}}, \boldsymbol{F}^{\text{ext}} \in \mathbb{R}^m$ and $\boldsymbol{M}_{\text{adm}}, \boldsymbol{D}_{\text{adm}} \in \mathbb{R}^{m \times m}$. It is thus important to note, that the virtual inertia and damping matrices, $\boldsymbol{M}_{\text{adm}}$ and $\boldsymbol{D}_{\text{adm}}$ respectively, do not represent a real physical system. However, they are hyperparameters reflecting the desired behavior of the system. The motion variables $\dot{\boldsymbol{x}}_{\text{des}}$ and $\ddot{\boldsymbol{x}}_{\text{des}}$ consequently express the motion of the system which possesses the desired damping and inertial properties, while being affected by the measured external wrenches. In order to transfer the desired motion to the real system, the resulting desired velocities are subsequently forwarded to a velocity controller. The latter, on the other hand, produces the required joint torques for the actuating motors. These, in turn, enable the real system to behave as the desired model.

3.4. Task Hierarchy

The concepts of task hierarchies and null space projectors [16–18] have already been introduced in Section 2.1. Given the complexity and importance of these concepts for

the remainder of this thesis, they are briefly recapitulated and covered in more detail throughout this section. This section is mainly inspired by the explanations provided by Dietrich et al. [9].

Task hierarchies are applicable for robots which possess more DOF than the number of variables needed to fulfill a particular task. These robots are said to be kinematically redundant. The surplus of DOF can hence be used to pursue additional tasks simultaneously. Expressed differently, a robot with a joint vector $\boldsymbol{q} \in \mathbb{R}^n$ is said to be kinematically redundant with respect to a task $\boldsymbol{x} \in \mathbb{R}^m$ if $n > m$. In this case, the robot is said to possess a kinematic redundancy of *n-m* [8, 9].

However, simply super-positioning multiple joint torques responsible for a range of tasks would cause the different tasks to compete against each other [9]. In order to avoid this, a task hierarchy is defined. For the purpose of establishing aforementioned hierarchy, null space projectors are introduced to instill priority levels for the individual tasks. In the following, the concepts of null space projectors and task hierarchies are presented for the example of two tasks with different priority levels. Afterwards, the observations are generalized for the case of an arbitrary number of simultaneous tasks.

Assume a robot with the joint vector $\boldsymbol{q} \in \mathbb{R}^n$. Two tasks defined by the operational coordinate positions $\boldsymbol{x}_1 \in \mathbb{R}^{m_1}$ and $\boldsymbol{x}_2 \in \mathbb{R}^{m_2}$ are defined. Their derivatives with respect to time are given by

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{J}_1(\boldsymbol{q})\dot{\boldsymbol{q}} \quad \dot{\boldsymbol{x}}_2 = \boldsymbol{J}_2(\boldsymbol{q})\dot{\boldsymbol{q}} \quad . \tag{3.22}$$

The two joint torque vectors $\boldsymbol{\tau}_1 \in \mathbb{R}^n$ and $\boldsymbol{\tau}_2 \in \mathbb{R}^n$ were computed separately in order to fulfill their respective tasks. What is more, one must define a prioritization where the highest priority is attributed to the first and the lower priority to the second task. Using the aforementioned null space projectors enables the projection of the second task the null space of the first task. Thus, the task of highest priority should be pursued with the robot's entire capabilities and not be hindered by conflicts that may arise with the second task. Put differently, a task of a lower priority should only be pursued if it is not disturbing any tasks of higher priority [5].

The projection $\boldsymbol{\tau}_2^P$ of joint torque $\boldsymbol{\tau}_2$ into the null space of the first task is calculated as

$$\boldsymbol{\tau}_2^P = \boldsymbol{N}_2^{\text{suc}}(\boldsymbol{q})\boldsymbol{\tau}_2 \in \mathbb{R}^n \quad , \tag{3.23}$$

with the "dynamically consistent" [21] null space projector of the second task $\boldsymbol{N}_2^{\text{suc}}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$. The term "dynamically consistent" refers to the joint torques linked to the null space projector not causing accelerations for the higher prioritized tasks inside the operational space [21]. It should thus be noted, that the inverse is not fulfilled. That is, a higher prioritized task can cause accelerations on a task of lower priority. According to the definition provided by Khatib [20], the null space projector is hence defined as

$$\boldsymbol{N}_2^{\text{suc}}(\boldsymbol{q}) = \boldsymbol{I} - \boldsymbol{J}_1(\boldsymbol{q})^T(\boldsymbol{J}_1(\boldsymbol{q})^{\boldsymbol{M}+})^T \quad . \tag{3.24}$$

Let $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ denote the identity matrix and $\boldsymbol{J}_1(\boldsymbol{q})^{\boldsymbol{M}+}$ the generalized inverse weighted by the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$. What is more, let the latter be defined as

$$\boldsymbol{J}_1(\boldsymbol{q})^{M+} = \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}_1(\boldsymbol{q})^T(\boldsymbol{J}_1(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{J}_1(\boldsymbol{q})^T)^{-1} \quad . \tag{3.25}$$

With the projected torque of the second task, the combined torque command applied on the robot is given by

$$\boldsymbol{\tau} = \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2^P \quad . \tag{3.26}$$

The proof of consistency for the defined null space projector can thus be found in Section B.3.

In the following, the presented case of two tasks is extended to an arbitrary number of tasks r. Hence, this can be expressed by the vectors

$$\boldsymbol{x}_i \in \mathbb{R}^{m_i}, \tag{3.27}$$

for $1 \leq i \leq r$ with m_i being the dimension of the ith task's coordinates. The lower the index of the task, the higher the priority. Therefore, $i = 1$ represents the task with the highest and $i = r$ the task with the lowest priority. The velocity in the task space is then calculated using the Jacobian $\boldsymbol{J}_i(\boldsymbol{q}) \in \mathbb{R}^{m_i \times n}$. For the ith task, the velocity is calculated by

$$\dot{\boldsymbol{x}}_i = \boldsymbol{J}_i(\boldsymbol{q})\dot{\boldsymbol{q}} \quad . \tag{3.28}$$

The multitude of tasks provides a new requirement for the projection: a lower prioritized task $\boldsymbol{\tau}_i$ is not only projected into the null space of the next highest prioritized task $\boldsymbol{\tau}_{i-1}$. Instead, it is projected into the null space of all higher prioritized tasks. This can be accomplished with the use of augmented variables [49]. In this sense, it is proceeded analogously to the previously covered case differing only by the use of the augmented Jacobian instead of the Jacobian matrix. This matrix stacks all of the equally or higher prioritized task related Jacobian matrices in to one single matrix. This augmented Jacobian is defined as

$$\boldsymbol{J}_{i-1}^{\text{aug}}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_1(\boldsymbol{q}) \\ \boldsymbol{J}_2(\boldsymbol{q}) \\ \vdots \\ \boldsymbol{J}_{i-1}(\boldsymbol{q}) \end{bmatrix} \quad . \tag{3.29}$$

The ith augmented null space projector is then calculated by

$$\boldsymbol{N}_i^{\text{aug}}(\boldsymbol{q}) = \boldsymbol{I} - \boldsymbol{J}_{i-1}^{\text{aug}}(\boldsymbol{q})^T(\boldsymbol{J}_{i-1}^{\text{aug}}(\boldsymbol{q})^{M+})^T \quad . \tag{3.30}$$

In order to reduce the computational effort, the augmented null space projector can alternatively be calculated recursively [22, 49, 50]. The recursive calculation is given by

$$\boldsymbol{N}_1^{\text{aug}} = \boldsymbol{I} \tag{3.31}$$

$$\hat{\boldsymbol{J}}_i(\boldsymbol{q}) = \boldsymbol{J}_i(\boldsymbol{q})\boldsymbol{N}_i^{\text{aug}}(\boldsymbol{q})^T \tag{3.32}$$

$$\boldsymbol{N}_i^{\text{aug}} = \boldsymbol{N}_{i-1}^{\text{aug}}(\boldsymbol{I} - \hat{\boldsymbol{J}}_{i-1}(\boldsymbol{q})^T(\hat{\boldsymbol{J}}_{i-1}(\boldsymbol{q})^{M+})^T) \quad . \tag{3.33}$$

It should be noted, that the first null space projector $\boldsymbol{N}_1^{\text{aug}}$ is formulated for the sake of consistency only. As the primary task should be undisturbed, there is no null space for it to be projected into. Thus, the null space projector is equal to the identity matrix.

The projected ith joint torque vector is computed analogously to (3.23), namely as

$$\boldsymbol{\tau}_i^P = \boldsymbol{N}_i^{\text{aug}}(\boldsymbol{q})\boldsymbol{\tau}_i \quad . \tag{3.34}$$

The resulting total joint torque is given by

$$\boldsymbol{\tau} = \boldsymbol{\tau}_1 + \sum_{i=2}^{r} \boldsymbol{\tau}_i^P \tag{3.35}$$

and is to be applied on the joints.

4. The Physical Model

As mentioned in Subsection 2.3.2, the currently existing prototype of the SwarmRail system serves the purpose of demonstrating the mobility of the mobile unit only. What is more, it is not capable of integrating an additional fully functional robotic arm. Therefore, the foreseen whole-body controller is implemented and validated entirely in a physical simulation of the real-world system. Hence, this chapter evolves around the modeled system with all its underlying simplifications and assumptions.

At first, general assumptions regarding the joints, motors and physical effects are introduced. Secondly, the model of the mobile unit is introduced, placing the focus on its DOF. In a following step, the modeled rail-network is presented, with a special emphasis on the ability to center the system through the peculiar shape of the rail profiles. Building upon this, a simplification of the model is proposed, using the singular perturbation approach with a consequent proof of its applicability. In a final step, the two scenarios which summarize the robot's entire area of application treated throughout this thesis are presented.

4.1. General Assumptions

This section summarizes the essential assumptions underlying the physical simulation in which the whole-body control is implemented and tested. Hence, this section contains the main elements by which the simulated and real environments differ from one another.

Assumption 1: All dynamic and kinematic parameters of the robot can be identified correctly.

Throughout the simulation, we assume all physical parameters to be exactly known. Oppositely to real-world applications, unavoidable measurement errors are hence not considered. Accordingly, this thesis assumes flawless and noise-free sensor readings of positions, velocities, accelerations, forces and torques, respectively.

Assumption 2: Electrical dynamics are sufficiently fast to be neglected.

As mentioned in Section 3.2, joints are operated by the application of torques through the usage of motors. The underlying motors for both, the mobile unit and the robotic arm attached to it, are electric. Consequently, the input of these motors are voltages, hence generating currents which are proportional to the individual joint torques. What

is more, we assume the dynamics of the electrical relationship between voltages and currents to be considerably faster compared to the dynamics of the mechanical system. As a consequence, motors are considered as ideal torque sources [44, 51].

Assumption 3: Time delays are not considered.

It is assumed, that commanded torques on the motors are immediately transferred to the mechanical joints. Further, the availability of measurements, such as joint positions and velocities, are assumed to be immediate. Put differently, communication delays impose no further challenge.

Assumption 4: The underlying joint dynamics are not considered.

The justification underlying this assumption is adopted from the explanation given by Dietrich [5]. The apparent link inertia is hereby down-scaled using a fast time-scale inner torque controller inside a slow time-scale rigid body [52, 53]. Assuming large joint stiffness, the singular perturbation approach can be applied to justify the omission of the underlying dynamics between motors and links [44, 51, 54].

Assumption 5: Friction is neglected.

Once the newly-implemented controller is applied to a real system, the vector containing external wrenches applied on the joints would also include the effects of friction on the robot. However, for the scope of this thesis, friction is not explicitly modeled.

Assumption 6: All links and joints of the robot are rigid bodies.

Deformations of the bodies are assumed to be non-existent. Therefore, the motion of the robot is entirely described by the positions of the joints as well as the examined sudden vertical translation occurring at crossing points. The latter is further analyzed in Subsection 4.4.2.

Having established the general assumptions regarding the modeled physical effects, an explanation regarding the modeled mobile unit is provided in the following.

4.2. Mobile Unit

As addressed in Subsection 2.3.2, the robots implemented inside the SwarmRail system are a combination of two subsystems: the mobile unit and the LWR attached at its bottom. Although the set of eight omni-wheels would allow the robotic system to perform horizontal translations in arbitrary directions and a rotation about the vertical axis, in practice the underlying rail-network sets physical limits upon these possibilities. At crossing points, the mobile unit is capable of translations in two orthogonal directions or rotations about the vertical axis.

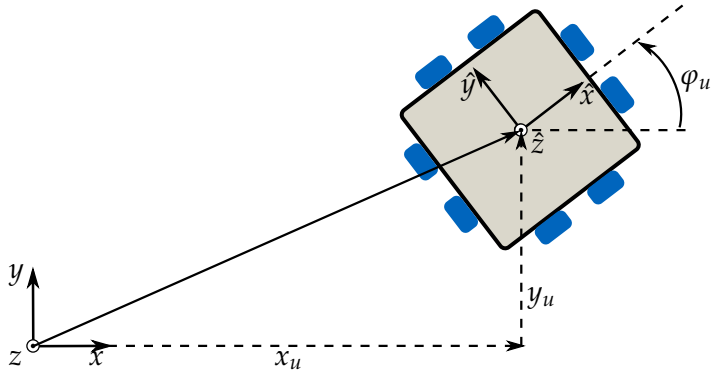


Figure 4.1.: Visualization of the world- and body-frame, along with the three kinematic values x_u, y_u and φ_u which are able to describe the mobile unit's whole range of motion.

An alternative description can be obtained through the comparison of coordinate frames. Therefore, a stationary reference frame O-xyz is defined, which is henceforth referred to as the "world-frame". The frame is oriented in such a way, that the x- and y-axes are parallel to the two possible translation directions inside the rail-network. The z-axis is vertical, having its direction oppose gravity. This frame definition is used for the remainder of this thesis in order to describe wrenches, positions, orientations, velocities and accelerations. The mobile unit is described by a body-frame frame $\hat{O}$-$\hat{x}\hat{y}\hat{z}$, located at the mobile unit's Center of Mass (COM) with the $\hat{z}$-axis being parallel to z. $\hat{x}$ and $\hat{y}$ are chosen as orthogonal to the edges of the mobile unit (refer to Figure 4.1).

At the crossing points, the robot is capable of performing translations in x and y, as well as rotations about the vertical axis z. Once the robot's mobile unit enters a straight segment, the set of possible motions is reduced to either a translation in x or in y, depending on the direction of the rails. All motions are hence expressed through the values x_u, y_u and φ_u, representing the translation along the x and y axes, as well as the rotation about z, respectively (refer to Figure 4.1).

Despite possessing four actuating elements, the mobile unit's movement is limited to the previously introduced three independent movements only. Hence, another assumption is made:

Assumption 7: The motions in the Cartesian directions are assumed to be the DOF of the mobile unit.

Henceforth, it is assumed that the implemented controller operates with torques and velocities acting on all of the three Cartesian directions. Consequently, we extend the assumption by adding an underlying kinematic controller. This is hence capable of translating the desired motions from the Cartesian space to commands for the individual motors connected to the omni-wheels, respectively.

These new DOF of the mobile unit can then be summarized by the vector $\hat{\boldsymbol{r}}$, which is defined as

$$\hat{\boldsymbol{r}} = \begin{bmatrix} y_u \\ \varphi_u \\ x_u \end{bmatrix} \in \mathbb{R}^3 \quad . \tag{4.1}$$

The peculiar order of the motion variables serves the purpose of separating constrained DOF from unconstrained DOF. This property becomes useful at the end of this section.

The respective translations and rotations are modeled as independent from one another. Implicitly, this introduces an additional assumption:

Assumption 8: The contact and driving forces of opposing wheels are assumed to be interminably symmetrical.

Experience has shown, that the mobile unit is sensitive to manufacturing and construction inaccuracies of the two L-shaped profiles, which together make up the straight segments the robot moves on (refer to Section 2.4). If both profiles are not perfectly aligned in one plane, the contact forces between opposing wheels become asymmetric. Therefore, the mobile unit is affected by varying driving forces at opposing sides. Consequently, the robot shows an undesired minor rotation once it is meant to drive in a straight line. While the physical borders of the rails prevent significant orientation changes inside the straight segments, they become evident once the robot enters a crossing point. In these positions, the robot is not cornered by any of the aforementioned borders. Put differently, the robot is free to rotate even further. Nevertheless, these complications do not receive any further consideration in this thesis. Therefore, all of the mobile unit's DOF can be assumed to be independent from one another

The mobile unit's symmetric and positive definite inertia matrix $\boldsymbol{M}_r \in \mathbb{R}^{3\times3}$ is calculated using the mass defined in Table 2.1. The mass of the mobile unit shall further be denoted by $m_{\text{unit}} = 17.5\text{kg}$.

The LWR attached to the robot is assumed to be a LWR IV, and possesses seven rotational DOF. Hence, each element q_i corresponds to the angle of the respective arm's ith joint. These DOF can hence be expressed by the vector $\boldsymbol{q}$, which is defined as

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_7 \end{bmatrix} \in \mathbb{R}^7 \quad . \tag{4.2}$$

The LWR has a mass of $m_{\text{LWR}} = 11.8\text{kg}$, with the respective symmetric and positive definite inertia matrix being defined as $\boldsymbol{M}_A \in \mathbb{R}^{7\times7}$.

Combining both systems leads to a system with ten DOF in total, which can be described by the vector

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{\boldsymbol{r}} \\ \boldsymbol{q} \end{bmatrix} \in \mathbb{R}^{10} \quad . \tag{4.3}$$

The total mass of the system is given by

$$m_{\text{Tot}} = m_{\text{unit}} + m_{\text{LWR}} = 29.3\text{kg} \quad . \tag{4.4}$$

The resulting inertia matrix for the coupled system is defined by the foregoing matrices of the individual subsystems $\boldsymbol{M}_r$ and $\boldsymbol{M}_A$, as well as the inertia coupling matrix $\boldsymbol{M}_{r,A} \in \mathbb{R}^{3\times7}$. Using the property of symmetry for inertia matrices, the resulting total inertia matrix $\boldsymbol{M}_{\text{Tot}}(\hat{\boldsymbol{y}}) \in \mathbb{R}^{10\times10}$ gained through the combination of both systems, can be expressed as

$$\boldsymbol{M}_{\text{Tot}}(\hat{\boldsymbol{y}}) = \begin{bmatrix} \boldsymbol{M}_r & \boldsymbol{M}_{r,A} \\ \boldsymbol{M}_{r,A}^T & \boldsymbol{M}_A \end{bmatrix} \quad . \tag{4.5}$$

4.3. Rail-Network

As expressed in Subsection 2.3.1, the rail-network examined throughout this thesis corresponds to a rectangular structure. The shape defined for the network sets kinematic constraints which guide the robot in its movement by twofold means: firstly, for the straight sections of the rails, the mobile unit is unable to perform neither rotations about the z-axis nor translations orthogonal to the direction set by the straight segments. Secondly, the robot is prevented from falling from the ceiling. These kinematic constraints are modeled by forces exerted through three spring-dampers acting on the robot's COM. These spring-dampers' forces prevent the deviation of the robot's motion from the path set by the rails. It is important to note that the spring-damper acts solely on the mobile unit. The robotic arm is not directly affected by the exerted force. In the remainder of this section, the focus is set on the spring-dampers which "block" DOF.

Given the rail's high material strength, the spring constants used are chosen accordingly in order to assume values which exceed the ones of the virtual springs, used for the impedance-control, by far. The latter is introduced in Chapter 5. Choosing high values for bespoken gains leads to high wrenches directing the robot to the center of the rail. Potential deviations caused by the motors or external influences acting on the robot are immediately counteracted, pushing the robot to the center anew. Therefore, inside the straight segments the rails impose a constant "block" on two of the mobile unit's DOF despite the robot theoretically possessing ten DOF. Hence, in reality, at straight segments of the rail-network, the controller has unrestricted access to only eight DOF. Accordingly, this work proposes to reduce the vector of the DOF given by (4.3) to a vector with only eight, instead of ten DOF. This reduction is accomplished by eliminating both of the DOF in the restricted directions. This is hence justified, should the velocity and change of position of these DOF be negligible. Via the incorporation of the singular perturbation approach we are thus able to prove the feasibility of bespoken simplification.

Kokotovic and Sannuti [55–57] belonged to the first group of scientist to explore the application of singular perturbation theory for purpose of control. This approach is

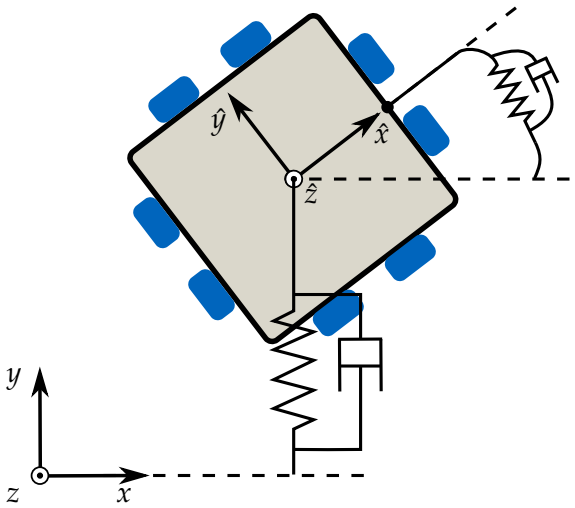


Figure 4.2.: Visualization of the two spring-damper systems which are used to model the kinematic constraint in the direction of the "blocked" DOF caused by the borders of the rails. This illustration thus replicates the case of the robot moving freely along the x-axis.

thus used for applications which can be described by twofold processes: hereby, one of these processes possesses considerably faster dynamics than the other, respectively. This circumstance motivates the separation of the entire system into two different time domains, reflecting the underlying individual dynamics. Although the number of examined systems increases, the complexity of each problem is reduced once compared to the original problem [58].

Motivated by this, the singular perturbation approach is implemented in order to investigate the kinematic constraints set by the rails. In the remainder of this section, the case of the robot moving in x-direction is covered as an example. Consequently, the restricted directions are the translation in y-direction as well as the rotation about the z-axis (refer to Figure 4.2). However, the results obtained with this example can be analogously transferred for the case of the mobile unit driving along the y-axis or rotating about the z-axis, whereas both remaining DOF remain "blocked".

In order to simulate the kinematic constraints set by the rail border, we choose the stiffness matrix for the "blocked" direction to be

$$\boldsymbol{K}_B = \begin{bmatrix} k_{B,1} & 0 \\ 0 & k_{B,2} \end{bmatrix} = \frac{1}{\epsilon^2} \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} = \frac{1}{\epsilon^2} \boldsymbol{K}_0 \quad , \tag{4.6}$$

for values satisfying $\epsilon, k_1, k_2 > 0$. With the condition $\epsilon \to 0$, the values for the stiffness matrix $\boldsymbol{K_B}$ are chosen to be very high, inducing dynamics that are much faster than the ones caused by the motors acting on the robotic system. The use of the singular perturbation approach and the associated introduction of two different time domains is therefore justified.

The associated damping values are chosen to be dependent on the stiffness affecting the corresponding joint value. The definition of the aforementioned damping values is inspired by the damping value for the damped harmonic one-dimensional oscillator [59]. They are thus defined as

$$d_{B,i} = D \cdot 2\sqrt{m_i k_{B,i}} = \underbrace{D \cdot 2\sqrt{m_i k_i}}_{d_i} \frac{1}{\epsilon} \tag{4.7}$$

for $i = 1, 2$. Let m_i represent the mass or inertia of the mobile unit with respect to the $\hat{z}$-axis, depending on whether the mobile unit's ith DOF causes either a translation or a rotation. Further, D represents the damping ratio of the system. Setting $D = 1$ creates a critically damped system. This choice enables the system to return to its equilibrium as fast as possible after an initial deviation, without causing oscillations [59].

Based on this information, the diagonal damping matrix $\boldsymbol{D}_B$ can be introduced, which is defined as

$$\boldsymbol{D}_B = \begin{bmatrix} d_{B,1} & 0 \\ 0 & d_{B,2} \end{bmatrix} = \frac{1}{\epsilon} \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} = \frac{1}{\epsilon} \boldsymbol{D}_0 \quad , \tag{4.8}$$

for $d_1, d_2 > 0$. In the case of unrestricted translation in the x-direction, the effect of the spring-damper on the entire mobile unit can be expressed by the joint torque vector

$$\boldsymbol{\tau}_R = -\frac{1}{\epsilon} \boldsymbol{D} \dot{\hat{\boldsymbol{y}}} - \frac{1}{\epsilon^2} \boldsymbol{K} \boldsymbol{\Delta} \hat{\boldsymbol{y}} \quad , \tag{4.9}$$

with $\boldsymbol{\Delta}\hat{\boldsymbol{y}}$ being a variation of the ten dimensional joint vector. It differs from the original vector (4.3) solely for both "blocked" DOF. These elements are replaced by the respective deviations from their equilibrium for the implemented spring. Thus, the vector is expressed as

$$\boldsymbol{\Delta}\hat{\boldsymbol{y}} = \begin{bmatrix} \Delta y_u \\ \Delta \varphi_u \\ x_u \\ q_1 \\ \vdots \\ q_7 \end{bmatrix} \quad . \tag{4.10}$$

Let Δ denote the deviation from the equilibrium pose, which is defined by the constant values $y_{u,0}$ and $\varphi_{u,0}$. Therefore, the first two elements of the vector $\boldsymbol{\Delta}\hat{\boldsymbol{y}}$ are given by

$$\begin{bmatrix} \Delta y_u \\ \Delta \varphi_u \end{bmatrix} = \begin{bmatrix} y_u - y_{u,0} \\ \varphi_u - \varphi_{u,0} \end{bmatrix} \quad . \tag{4.11}$$

The value $y_{u,0}$ denotes a robot located at the center of the rail while $\varphi_{u,0}$ corresponds to an orientation aligning $\hat{x}$ with the x-direction set by the world-frame (refer to Figure 4.2).

What is more, $\boldsymbol{D}, \boldsymbol{K} \in \mathbb{R}^{10\times 10}$ are chosen to be the symmetric matrices

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}_0 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_0 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \quad . \tag{4.12}$$

Due to their sparsity, these matrices only cause wrenches on the two restricted DOF. This property becomes relevant later in this section.

4.3.1. Time Domains

As previously explained, the high stiffness of the railing border introduces dynamics to the system which exceed the motions caused by external and motor torques, in terms of velocity, by far. Hence, this serves as a justification to distinct between two different time domains, typically implied in the field of singular perturbation approach [44]. The time variable t represent the slow time-scale of the system, which corresponds to the motions caused by motor torques or external disturbances. The time variable v represents the fast time-scale. This is hence influenced by the spring-damper systems representing the physical rail boundaries.

The derivatives referencing the slow time-scale variable t are denoted by dots ($\dot{}$). Meanwhile the derivatives referencing the fast time-scale variable v are denoted by prime ($'$).

The transition from the derivatives of the different time-scales can be expressed by the parameter ϵ as

$$\epsilon \frac{df}{dt} = \frac{df}{dv} = f' \tag{4.13}$$

and

$$\epsilon^2 \frac{d^2 f}{dt^2} = \frac{d^2 f}{dv^2} = f'' \quad . \tag{4.14}$$

Based on this idea, the joint variables of the robot can also be split into two separate parts: the first part, characterized by a bar, symbolizes the slow time-scale. The second part, denoted by the index v, represents the fast time-scale. This separation influences all DOF, which are thus expressed as

- $y_u = \overline{y}_u + y_{u,v}$
- $\varphi_u = \overline{\varphi}_u + x_{u,v}$
- $x_u = \overline{x}_u + x_{u,v}$
- $q_i = \overline{q}_i + q_{i,v}$

for $i \in [1,7]$.

Due to the fact of the domain v having considerably faster dynamics, the slow variables are assumed to remain constant when viewed from the fast time domain's perspective.

Therefore, when derivatives with respect to v are observed, only the parts of the joints representing the fast time-scale are considered. For the example of q_i this yields

$$q_i' = q_{i,v}' \quad . \tag{4.15}$$

In the following subsections the robotic system in investigated with respect to both: the slow and fast time domains.

4.3.2. Slow Dynamics

Before the implementation of the chosen control method, the EOM for the mobile unit can be expressed as

$$\boldsymbol{M}_{\text{Tot}}(\hat{\boldsymbol{y}})\ddot{\hat{\boldsymbol{y}}} + \boldsymbol{C}_{\text{Tot}}(\hat{\boldsymbol{y}}, \dot{\hat{\boldsymbol{y}}})\dot{\hat{\boldsymbol{y}}} + \hat{\boldsymbol{g}}(\hat{\boldsymbol{y}}) = \hat{\boldsymbol{\tau}}^{\text{ext}} + \boldsymbol{\tau}_R \tag{4.16}$$

$$\boldsymbol{M}_{\text{Tot}}(\hat{\boldsymbol{y}})\ddot{\hat{\boldsymbol{y}}} + \boldsymbol{C}_{\text{Tot}}(\hat{\boldsymbol{y}}, \dot{\hat{\boldsymbol{y}}})\dot{\hat{\boldsymbol{y}}} + \hat{\boldsymbol{g}}(\hat{\boldsymbol{y}}) + \frac{1}{\epsilon}\boldsymbol{D}\dot{\hat{\boldsymbol{y}}} + \frac{1}{\epsilon^2}\boldsymbol{K}\boldsymbol{\Delta}\hat{\boldsymbol{y}} = \hat{\boldsymbol{\tau}}^{\text{ext}} \quad . \tag{4.17}$$

Let $\hat{\boldsymbol{g}}(\hat{\boldsymbol{y}})$ denote the effects on the joints caused by gravity and $\boldsymbol{C}_{\text{Tot}}(\hat{\boldsymbol{y}}, \dot{\hat{\boldsymbol{y}}})$ denote the Coriolis matrix, both defined for the entire robotic system, i.e. the mobile unit with the attached LWR. Multiplying (4.17) with ϵ^2 results in

$$\epsilon^2\boldsymbol{M}_{\text{Tot}}(\hat{\boldsymbol{y}})\ddot{\hat{\boldsymbol{y}}} + \epsilon^2\boldsymbol{C}_{\text{Tot}}(\hat{\boldsymbol{y}}, \dot{\hat{\boldsymbol{y}}})\dot{\hat{\boldsymbol{y}}} + \epsilon^2\hat{\boldsymbol{g}}(\hat{\boldsymbol{y}}) + \epsilon\boldsymbol{D}\dot{\hat{\boldsymbol{y}}} + \boldsymbol{K}\boldsymbol{\Delta}\hat{\boldsymbol{y}} = \epsilon^2\hat{\boldsymbol{\tau}}^{\text{ext}} \quad . \tag{4.18}$$

By setting $\epsilon \to 0$, this equation is thus reduced to

$$\boldsymbol{K}\boldsymbol{\Delta}\hat{\boldsymbol{y}} = \boldsymbol{0} \quad . \tag{4.19}$$

The validity of this equation holds only, iff the joint values representing the "blocked" motions are equal to their respective equilibrium. Therefore, inside the slow time domain the mobile unit can be assumed to be located at the center of the rail, with x and $\hat{x}$ being parallel to each other (refer to Figure 4.2).

4.3.3. Fast Dynamics

Throughout this subsection the system described inside the fast time domain is examined. Therefore, the change of coordinates from t to v, as expressed by (4.13) and (4.14), is performed. Therefore, (4.18) can be reformulated as

$$\boldsymbol{M}_{\text{Tot}}(\hat{\boldsymbol{y}})\hat{\boldsymbol{y}}'' + \boldsymbol{C}_{\text{Tot}}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')\hat{\boldsymbol{y}}' + \boldsymbol{D}\hat{\boldsymbol{y}}' + \boldsymbol{K}\boldsymbol{\Delta}\hat{\boldsymbol{y}} = \boldsymbol{0} \quad , \tag{4.20}$$

with respect to the fast time-scale.

Based on (4.20), the energy-based storage function for the fast domain $S(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')$ can be defined as

$$S(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}') = \frac{1}{2}\hat{\boldsymbol{y}}'^{T}\boldsymbol{M}_{\text{Tot}}(\hat{\boldsymbol{y}})\hat{\boldsymbol{y}}' + \frac{1}{2}\boldsymbol{\Delta}\hat{\boldsymbol{y}}^{T}\boldsymbol{K}\boldsymbol{\Delta}\hat{\boldsymbol{y}} \quad . \tag{4.21}$$

Or expressed in matrix notation

$$S(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}') = \begin{bmatrix} \boldsymbol{\Delta}\hat{\boldsymbol{y}} \\ \hat{\boldsymbol{y}}' \end{bmatrix}^T \begin{bmatrix} \boldsymbol{K} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{\mathrm{Tot}}(\hat{\boldsymbol{y}}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}\hat{\boldsymbol{y}} \\ \hat{\boldsymbol{y}}' \end{bmatrix} \quad . \tag{4.22}$$

Due to the matrix $\boldsymbol{K}$ being positive semidefinite only (refer to Section A.4), the entire storage function is also positive semidefinite [60]. Analyzing the derivative of the storage function with respect to v results in

$$S(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')' = \frac{1}{2}\hat{\boldsymbol{y}}''^T \boldsymbol{M}_{\mathrm{Tot}}(\hat{\boldsymbol{y}})\hat{\boldsymbol{y}}' + \frac{1}{2}\hat{\boldsymbol{y}}'^T \boldsymbol{M}_{\mathrm{Tot}}(\hat{\boldsymbol{y}})\hat{\boldsymbol{y}}'' + \frac{1}{2}\hat{\boldsymbol{y}}'^T \boldsymbol{M}_{\mathrm{Tot}}(\hat{\boldsymbol{y}})'\hat{\boldsymbol{y}}' + \frac{1}{2}\hat{\boldsymbol{y}}'^T \boldsymbol{K}\boldsymbol{\Delta}\hat{\boldsymbol{y}} + \frac{1}{2}\boldsymbol{\Delta}\hat{\boldsymbol{y}}^T \boldsymbol{K}\hat{\boldsymbol{y}}' \quad . \tag{4.23}$$

Due to the symmetry of the matrices $\boldsymbol{M}_{\mathrm{Tot}}(\hat{\boldsymbol{y}})$ and $\boldsymbol{K}$ this set of equations can be reduced to

$$S(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')' = \hat{\boldsymbol{y}}'^T \boldsymbol{M}_{\mathrm{Tot}}(\hat{\boldsymbol{y}})\hat{\boldsymbol{y}}'' + \frac{1}{2}\hat{\boldsymbol{y}}'^T \boldsymbol{M}_{\mathrm{Tot}}(\hat{\boldsymbol{y}})'\hat{\boldsymbol{y}}' + \hat{\boldsymbol{y}}'^T \boldsymbol{K}\boldsymbol{\Delta}\hat{\boldsymbol{y}} \quad . \tag{4.24}$$

Using the relation (4.20) for $\boldsymbol{M}_{\mathrm{Tot}}(\hat{\boldsymbol{y}})\hat{\boldsymbol{y}}''$ yields

$$S(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')' = \hat{\boldsymbol{y}}'^T[-\boldsymbol{C}_{\mathrm{Tot}}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')\hat{\boldsymbol{y}}' - \boldsymbol{D}\hat{\boldsymbol{y}}' - \boldsymbol{K}\boldsymbol{\Delta}\hat{\boldsymbol{y}}] + \frac{1}{2}\hat{\boldsymbol{y}}'^T \boldsymbol{M}_{\mathrm{Tot}}(\hat{\boldsymbol{y}})'\hat{\boldsymbol{y}}' + \hat{\boldsymbol{y}}'^T \boldsymbol{K}\boldsymbol{\Delta}\hat{\boldsymbol{y}} \quad . \tag{4.25}$$

Using the properties (3.14) and (4.13), the latter equation can be formulated as

$$S(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')' = -\hat{\boldsymbol{y}}'^T \boldsymbol{D}\hat{\boldsymbol{y}}' + \frac{1}{2}\hat{\boldsymbol{y}}'^T \left(\boldsymbol{C}_{\mathrm{Tot}}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')^T - \boldsymbol{C}_{\mathrm{Tot}}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')\right)\hat{\boldsymbol{y}}' \quad . \tag{4.26}$$

Given the skew symmetry of $\boldsymbol{C}_{\mathrm{Tot}}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')^T - \boldsymbol{C}_{\mathrm{Tot}}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')$, the equation can be further simplified to

$$S(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')' = -\hat{\boldsymbol{y}}'^T \boldsymbol{D}\hat{\boldsymbol{y}}' \quad . \tag{4.27}$$

Due to the sparsity of the damping matrix $\boldsymbol{D}$, an additional simplification results in

$$S(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')' = -\begin{bmatrix} y_u' \\ \varphi_u' \end{bmatrix}^T \boldsymbol{D}_0 \begin{bmatrix} y_u' \\ \varphi_u' \end{bmatrix} = -\left(d_1(y_u')^2 + d_2(\varphi_u')^2\right) \leq 0 \quad . \tag{4.28}$$

The storage function is positive semidefinite while its derivative with respect to v is negative semidefinite. As a consequence, no proof regarding the stability of the system in the fast time can be expected whilst using the energy-based storage function. However, another relevant information can be obtained from the resulting equations. Since $S(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')'$ is negative semidefinite, the systems' energy decreases or remains stationary. Therefore, the initial state sets an upper boundary for the total energy of the system inside the fast time-scale. This however consequently implies, that the initial state also sets an upper boundary for the velocities of the robot. Further, assuming a robot

starting from a resting position, the initial values for the velocities in the v-domain can be assumed to be small at all times.

The aforementioned small and bounded values for the velocities also effect the EOM (see (4.20)). Hence, the definition of the Coriolis matrix (3.13), reveals that the Coriolis and centrifugal forces expressed by $\boldsymbol{C}_{\text{Tot}}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')\hat{\boldsymbol{y}}'$ possess a quadratic dependency with respect to the velocity. The approximation $\boldsymbol{C}_{\text{Tot}}(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}')\hat{\boldsymbol{y}}' \approx \boldsymbol{0}$ can thus be justified by the assumption of small magnitudes of said initial velocities.

Hence, this measure modifies the EOM expressed by (4.20) to

$$\boldsymbol{M}_{\text{Tot}}(\hat{\boldsymbol{y}})\hat{\boldsymbol{y}}'' + \boldsymbol{D}\hat{\boldsymbol{y}}' + \boldsymbol{K}\Delta\hat{\boldsymbol{y}} = \boldsymbol{0} \quad . \tag{4.29}$$

This equation has an important property: except for $\boldsymbol{M}_{\text{Tot}}(\hat{\boldsymbol{y}})\hat{\boldsymbol{y}}''$, all the remaining terms are solely affected by the motion in the direction of the two "blocked" DOF (refer to (4.12)). In order to reduce (4.29) to an expression of the two "blocked" DOF, the joint vector $\hat{\boldsymbol{y}}$ is separated into

$$\boldsymbol{y}_B = \begin{bmatrix} \Delta y_u \\ \Delta \varphi_u \end{bmatrix} \text{ and } \boldsymbol{y} = \begin{bmatrix} x_u \\ q_1 \\ q_2 \\ \vdots \\ q_7 \end{bmatrix} \quad \text{, respectively.} \tag{4.30}$$

Due to the chosen order for the mobile unit's motion variables combined in $\hat{\boldsymbol{r}}$, the vector $\hat{\boldsymbol{y}}$ can be viewed as the vertical stacking of the two restricted and the eight unrestricted DOF. Based on this, the symmetric inertia matrix can be decomposed into four block matrices, therefore being expressed by

$$\begin{bmatrix} \boldsymbol{M}_B(\boldsymbol{y}_B) & \boldsymbol{M}_{B,y}(\boldsymbol{y}_B, \boldsymbol{y}) \\ \boldsymbol{M}_{B,y}(\boldsymbol{y}_B, \boldsymbol{y})^T & \boldsymbol{M}_y(\boldsymbol{y}) \end{bmatrix} \hat{\boldsymbol{y}}'' + \begin{bmatrix} \boldsymbol{D}_0 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \hat{\boldsymbol{y}}' + \begin{bmatrix} \boldsymbol{K}_0 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \Delta\hat{\boldsymbol{y}} = \boldsymbol{0} \quad . \tag{4.31}$$

Let $\boldsymbol{M}_B(\boldsymbol{y}_B) \in \mathbb{R}^{2\times 2}$, $\boldsymbol{M}_y(\boldsymbol{y}) \in \mathbb{R}^{8\times 8}$ and $\boldsymbol{M}_{B,y}(\boldsymbol{y}_B, \boldsymbol{y}) \in \mathbb{R}^{2\times 8}$. Therefore (4.31) can be separated into

$$\boldsymbol{M}_B(\boldsymbol{y}_B)\boldsymbol{y}_B'' + \boldsymbol{M}_{B,y}(\boldsymbol{y}_B, \boldsymbol{y})\boldsymbol{y}'' + \boldsymbol{D}_0\boldsymbol{y}_B' + \boldsymbol{K}_0\boldsymbol{y}_B = \boldsymbol{0} \tag{4.32}$$

and

$$\boldsymbol{M}_{B,y}(\boldsymbol{y}_B, \boldsymbol{y})^T\boldsymbol{y}_B'' + \boldsymbol{M}_y(\boldsymbol{y})\boldsymbol{y}'' = \boldsymbol{0} \quad . \tag{4.33}$$

Imposing (4.33), $\boldsymbol{y}''$ can be expressed in dependence of $\boldsymbol{y}_B''$ by

$$\boldsymbol{y}'' = -\boldsymbol{M}_y(\boldsymbol{y})^{-1}\boldsymbol{M}_{B,y}(\boldsymbol{y}_B, \boldsymbol{y})^T\boldsymbol{y}_B'' \quad . \tag{4.34}$$

Using (4.32) leads to

$$\boldsymbol{M}_{\text{red}}(\hat{\boldsymbol{y}})\boldsymbol{y}_B'' + \boldsymbol{D}_0\boldsymbol{y}_B' + \boldsymbol{K}_0\boldsymbol{y}_B = \boldsymbol{0} \quad , \tag{4.35}$$

with:

$$\boldsymbol{M}_{\text{red}}(\hat{\boldsymbol{y}}) = (\boldsymbol{M}_B(\boldsymbol{y}_B) - \boldsymbol{M}_{B,y}(\boldsymbol{y}_B, \boldsymbol{y})\boldsymbol{M}_y(\boldsymbol{y})^{-1}\boldsymbol{M}_{B,y}(\boldsymbol{y}_B, \boldsymbol{y})^T) \quad . \tag{4.36}$$

The new reduced inertia matrix $\boldsymbol{M}_{\text{red}}$ is positive definite. This is verifiable by the Schur complement conditions [61] (refer to Section A.6), which are fulfilled since $\boldsymbol{M}_{\text{Tot}}(\hat{\boldsymbol{y}})$ and $\boldsymbol{M}_y(\boldsymbol{y})$ are positive definite.

The new reduced EOM can now be examined with respect to the stability, by adopting the "modified Lyapunov", as proposed by Whitcomb et al. [62]. With $\gamma \to 0$, the storage function is defined as

$$S_B(\boldsymbol{y}_B, \boldsymbol{y}_B') = \frac{1}{2}\boldsymbol{y}_B'^T\boldsymbol{M}_{\text{red}}(\hat{\boldsymbol{y}})\boldsymbol{y}_B' + \frac{1}{2}\boldsymbol{y}_B^T\boldsymbol{K}_0\boldsymbol{y}_B + \gamma \cdot \boldsymbol{y}_B'^T\boldsymbol{M}_{\text{red}}(\hat{\boldsymbol{y}})\boldsymbol{y}_B \quad , \tag{4.37}$$

or alternatively expressed as a matrix product

$$S_B(\boldsymbol{y_B}, \boldsymbol{y}_B') = \frac{1}{2}\begin{bmatrix}\boldsymbol{y}_B \\ \boldsymbol{y}_B'\end{bmatrix}^T \begin{bmatrix}\boldsymbol{K}_0 & \gamma\boldsymbol{M}_{\text{red}}(\hat{\boldsymbol{y}}) \\ \gamma\boldsymbol{M}_{\text{red}}(\hat{\boldsymbol{y}})^T & \boldsymbol{M}_{\text{red}}(\hat{\boldsymbol{y}})\end{bmatrix}\begin{bmatrix}\boldsymbol{y}_B \\ \boldsymbol{y}_B'\end{bmatrix} \quad . \tag{4.38}$$

Deriving this modified storage function by v leads to:

$$S_B(\boldsymbol{y}_B, \boldsymbol{y}_B')' = -\begin{bmatrix}\boldsymbol{y}_B \\ \boldsymbol{y}_B'\end{bmatrix}^T \begin{bmatrix}\gamma\boldsymbol{K}_0 & \frac{\gamma}{2}\boldsymbol{D}_0 \\ \frac{\gamma}{2}\boldsymbol{D}_0^T & \boldsymbol{D}_0 - \gamma\boldsymbol{M}_{\text{red}}(\hat{\boldsymbol{y}})\end{bmatrix}\begin{bmatrix}\boldsymbol{y}_B \\ \boldsymbol{y}_B'\end{bmatrix} \quad . \tag{4.39}$$

Whitcomb et al. [62] prove, that the defined storage function $S_B(\boldsymbol{y_B}, \boldsymbol{y}_B')$ is positive definite and its derivative, with respect to v, namely $S_B(\boldsymbol{y}_B, \boldsymbol{y}_B')'$, is negative definite, for a sufficiently small value for γ. This requirement is fulfilled, since $\gamma \to 0$ is assumed.

Further, referencing to the work of Murray et al. [41], we are able to conclude exponential stability towards the point of equilibrium inside the fast time-scale. The displacements and velocities in the "blocked" directions are not only bounded to be small inside the fast time domain, they additionally decay exponentially to their equilibrium. Expressed differently: the system converges exponentially to the central position inside the rails.

This allows a further assumption for the mobile unit:

Assumption 9: The mobile unit is assumed to be centered with respect to the rails at all times.

Further, the EOM can be simplified to a system with only eight DOF, omitting the two "blocked" directions. Expressed differently: modeling the rail-network as a stiff spring-damper system allows to reduce the EOM to being dependant of the unrestricted DOF contained in $\boldsymbol{y}$ only.

This also has a consequence for the unrestricted DOF. Assuming the restricted DOF to be at their respective equilibrium, the corresponding spring-dampers modeling the rail borders do not impart any kind of wrenches on the system. As the high valued spring-dampers were the source of the fast dynamics to begin with, we can further

assume the dynamics of the system to be fully described by the slow dynamics. Put differently, a fast time-scale component can not be established in the unrestricted DOF.

As stated at the very beginning of this section, although the derivations were performed for a mobile unit moving along the x-axis, the results' applicability is not limited to solely this case. This can be easily verified, by modifying the order of the mobile unit's joint values in $\hat{\boldsymbol{r}}$ (refer to (4.1)). Therefore, the new variable $r \in \mathbb{R}$ is introduced, which contains the mobile unit's unrestricted DOF. This value can correspond to the translation in x or y, as well as the rotation about the vertical axis, depending on the current location of the robot. Therefore, we can generalize $\boldsymbol{y}$ as a combination between r and the unmodified DOF of the LWR. This results in

$$\boldsymbol{y} = \begin{bmatrix} r \\ \boldsymbol{q} \end{bmatrix} \in \mathbb{R}^8 \quad . \tag{4.40}$$

Consequently, the inertia matrix $\boldsymbol{M}_y$ can thus also be expressed with respect to the mobile unit's generalized unrestricted DOF. This representation is given by

$$\boldsymbol{M}_y = \begin{bmatrix} m_r(r) & \boldsymbol{m}_{r,A}(\boldsymbol{y}) \\ \boldsymbol{m}_{r,A}(\boldsymbol{y})^T & \boldsymbol{M}_A(\boldsymbol{q}) \end{bmatrix} \quad , \tag{4.41}$$

with $m_r \in \mathbb{R}$ being either the mass or the moment of inertia of the mobile unit with respect to the $\hat{z}$-axis, depending on whether a translation or rotation is allowed. Nevertheless, $\boldsymbol{m}_{r,A}(\boldsymbol{y}) \in \mathbb{R}^{1\times 7}$ represents the inertia coupling term between the mobile unit and the LWR. Analogously, the Coriolis matrix is divided in the same manner, resulting in the EOM

$$\begin{bmatrix} m_r(r) & \boldsymbol{m}_{r,A}(\boldsymbol{y}) \\ \boldsymbol{m}_{r,A}(\boldsymbol{y})^T & \boldsymbol{M}_A(\boldsymbol{q}) \end{bmatrix} \ddot{\boldsymbol{y}} + \begin{bmatrix} c_r(r,\dot{r}) & \boldsymbol{c}_{r,A}(\boldsymbol{y},\dot{\boldsymbol{y}}) \\ \boldsymbol{c}_{r,A}(\boldsymbol{y},\dot{\boldsymbol{y}})^T & \boldsymbol{C}_A(\boldsymbol{q},\dot{\boldsymbol{q}}) \end{bmatrix} \dot{\boldsymbol{y}} + \begin{bmatrix} g_r(\boldsymbol{y}) \\ \boldsymbol{g}_q(\boldsymbol{y}) \end{bmatrix} = \boldsymbol{\tau}^{\text{ext}} \quad . \tag{4.42}$$

4.4. Modeled Scenarios

In accordance with our choice of a rectangular grid structure, the motion of the mobile unit can be divided into two different scenarios, namely:

- Rotational Scenario (RS): The mobile unit is located at a crossing point and is able to either rotate or remain stationary.
- Translational Scenario (TS): The mobile unit drives along the rail-network unidirectionally.

Any motion performed inside the rail-network can be expressed as a combination sequence of these scenarios. However, this thesis treats both cases separately. Furthermore, the TS is limited to solely cover one translation direction at once. The non-trivial fluent transition between differing translational directions is disregarded for the scope of this thesis. Notwithstanding the previous limitations, the remaining two scenarios are described in the following subsections.

4.4.1. Rotational Scenario

Inside the RS, the mobile unit is restricted to either rotating or remaining stationary. Although the crossing point would theoretically enable the mobile unit to perform translations in four directions, the simulation of the RS does not take these translations into consideration. This can be traced back to the fact that performing a translation would cause a transition to the previously mentioned TS and therefore not add new insights to the controller's performance. Furthermore, the transition between two different scenarios would also imply a change in the "blocked" DOF (refer to Section 4.3). This procedure is not trivial and is thus not investigated throughout this thesis.

During the RS, the robot is assumed to be restrained by two spring-damper systems, acting in the x and y-axes of the mobile unit. Therefore, pushing the robot's COM to the center of the crossing at all times and only allowing a rotation about the z-axis. This measure prevents the mobile unit from performing translation, therefore transitioning to the TS.

4.4.2. Translational Scenario

As mentioned previously, the TS covers the one-directional translation of the mobile unit through the rail-network. During this scenario, the forces exerted on the robot by the railing border are assumed to act on the mobile unit at all times. Since this topic has already been discussed in extensive detail (please refer to Section 4.3) it is not given more attention in this subsection. However, this subsection aims to describe the manner of how the passage through a crossing point is modeled inside the simulation. These characteristic points represent a challenge for the controller implemented in the system, as it is demonstrated in Chapter 6.

Recapitulating the definition provided by Subsection 2.3.1, a crossing point is defined as the intersection of straight rail segments. When the mobile unit passes such a point, it transitions from one straight segment to the next. As outlined in Section 2.4, manufacturing and construction inaccuracies impose one problem for such crossing points: the two mentioned segments might differ in their respective altitude. The consequences of this circumstance are twofold: firstly, the entire robot experiences a sudden undesirable change in altitude. Secondly, the transition between consecutive segments is not smooth, but rather causes a collision between the unit's wheels and the subsequent rail segment.

The crossing points are modeled as locations at which forces appear or disappear, depending on the distance between the mobile unit and said crossing points. Following the rectangular rail dimensions, the crossing points are placed in a grid-structure separated by a distance of 1m along x and y. In order to model the altitude change, the vertical spring-damper is removed for a small time t_{Fall}. The described spring-damper removal takes place whenever the mobile unit's COM passes a natural multiple of the crossing point distance of 1m. Due to the vertical force being absent, the robot falls in the direction gravity acts upon. After falling for the predefined time t_{Fall}, the vertical

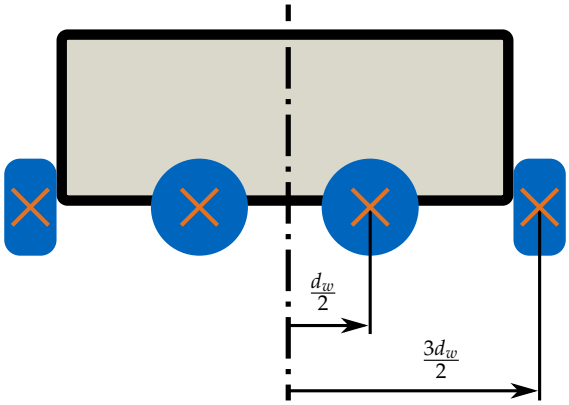


Figure 4.3.: Side view of the SwarmRail. The red crosses at the wheel's center points highlight the individual distances of the collision points, with respect to the center of the robot.

spring-damper is applied again. Thereby, a sudden upward force is introduced once more, pushing the robot back to its original position.

Additionally to the sudden position change, the varying altitude between consecutive rail segments has another consequence. When transitioning between consecutive straight segments, the wheels also present frontal collisions with the consequent straight segment. Therefore, each passing of a crossing point is accompanied by four separate collisions, one for each set of opposing wheel pairs. In experiments performed with the demonstrator at DLR, it could be shown, that the impact forces are able to considerably slow down the robot in its translational motion through a crossing point.

As illustrated in Figure 4.3, these collisions occur whenever the robot's center has a distance of either $\frac{d_w}{2}$ or $\frac{3d_w}{2}$ from the respective crossing points. Therefore, the full sequence for traversing a crossing point is given by two collisions at the distances $\frac{3d_w}{2}$ and $\frac{d_w}{2}$ between the center of mass and this particular crossing point. When the COM is located exactly at the crossing point, the mobile unit falls for a predetermined time t_{Fall}. The fall is followed by two more collisions at the distances $-\frac{d_w}{2}$ and $-\frac{3d_w}{2}$ between COM and crossing point. As illustrated in Figure 4.3, the distance d_w represents a third of the mobile unit's total length/width, i.e. d_w=120.7mm.

The impact forces depend on the unit's translation velocity, which is defined by $v_r \in \mathbb{R}$, the total mass of the robot m_{Tot} and the simulation's sample time Δt=1ms. Consequently, the horizontal impact force $F_{\text{impact}} \in \mathbb{R}$ acting on the COM in the direction of translation is calculated as

$$F_{\text{impact}} = -\frac{m_{\text{Tot}}}{\Delta t} v_r \quad . \tag{4.43}$$

Put differently: the force is selected sufficiently high such that a rigid body, presenting the mass and horizontal velocity of the mobile unit, can be brought to a stand still in the minimal amount of time defined by the simulation environment. That is, for a system

without friction, centrifugal forces or damping terms and with no further acting forces. Thus, the impact forces have been set particularly high, therefore reflecting the worst case possible. Although in reality smaller forces are to be expected, this model allows to test the controller's capability of reacting to intense and abrupt disturbances.

The simplified model of the collision impact is legitimized by the previously made assumption, stating that the robot is assumed to have equal symmetric driving forces (Assumption 8) and be centered when entering the crossing point (Assumption 9). Thus, the wheels at opposing sides collide simultaneously with the consequent rail segment at the same velocity. Therefore, the resulting sequence of impact forces can be assumed to act on the mobile unit's COM, i.e. no torques are caused.

This is a discrepancy between the simulation and the environment which can be expected to appear under real-world conditions. Should the mobile unit enter the crossing point in a skewed manner, the time of impact and magnitude of the collision force between wheels and the next straight segment differ for the opposing sides. Consequently, the mobile unit is exposed to an additional torque. This is, however, not investigated in this thesis.

It should also be thus stated, that under real-world conditions the entire process of crossing point passing occurs as one fluent phenomenon. The division into two independent events, namely a change in altitude of the COM and frontal collisions, is not fulfilled. Additionally, the altitude change does not take place as a sudden change in the vertical direction. Rather, the robot experiences a sequence of tilts about the horizontal axis, orthogonal to the direction of translation. These tilts are a byproduct for each time a set of wheels loses or gains contact to the rails, which occurs eight times per passed crossing point. For the purpose of this thesis, namely examining the performance of the whole-body impedance controller for sudden abrupt positional changes, however, the reduced complexity model suffices.

5. Control Method

This chapter introduces the structure of the SwarmRails' hierarchical whole-body impedance tracking controller, whose implementation is the central subject of this thesis. This controller represents a combination of the whole-body controllers developed at the DLR [5–7], applied inside the robotic systems Rollin' Justin [63] and KUKA LWR IV+. Combining the aforementioned whole-body controller concepts allows handling the nonholonomy of the mobile unit while enabling the execution of multiple tasks inside a strict hierarchy. By adopting a strict hierarchy, we are able to avoid tasks competing and consequently interfering with each other. Additionally, the natural inertia preserving approach [6] has proven to be robust against external forces, even without being able to measure them. This makes this controller a particularly promising potential controller for SwarmRail, which is heavily influenced by collisions.

These inspirational papers establish the division of this chapter into two sections. The first section follows the work of Dietrich [5, 7], being concerned with the connection between the kinematically-controlled mobile unit and the attached torque-controlled LWR. The second section builds upon these findings, expanding the controller to hierarchical impedance-based tracking control, as developed by Dietrich et al. [6].

5.1. Admittance Interface

A whole-body impedance controller operates based on force and torque control algorithms [64]. The mobile unit, however, introduces an additional complexity to the system. Due to the kinematic rolling constraints at the contact points between the mobile unit's wheels and the rails, the system becomes nonholonomic [65]. In order to, nevertheless, ensure consistent locomotion of the mobile unit, it must be resorted to kinematic control laws for this part of the robot. The implementation of such a controller, prevents the controller's access to the mobile unit via the typical force or torque interface [5].

The admittance-based control, which has already been introduced in Subsection 3.3.2, is an example for such a kinematic controller. However, since it is aimed to incorporate the mobile unit in a whole-body impedance controller the existence of a force-torque interface is indispensable [5]. This interface serves as the connection point between the mobile unit's admittance-based and the LWRs torque-based controller.

Considering the mobile unit's reduced physical model, i.e. assuming only one DOF (refer to Section 4.3), the EOM for the kinematically controlled mobile unit can be expressed as

$$m_{r,\mathrm{adm}}\ddot{r}_{\mathrm{des}} + d_{r,\mathrm{adm}}\dot{r}_{\mathrm{des}} = \tau_r + \tau_r^{\mathrm{ext}} \quad . \tag{5.1}$$

For the purpose of our controller we chose the virtual damping value to be equal to zero. This choice enables the derivation of the system's asymptotic stability for the applied hierarchical controller. The latter is further explained at the end of this section. All in all, the virtual mobile unit of our controller can be described by

$$m_{r,\mathrm{adm}}\ddot{r}_{\mathrm{des}} = \tau_r + \tau_r^{\mathrm{ext}} \quad . \tag{5.2}$$

Analogously to the vector expressed by (4.40), the mobile unit is analysed with respect to a generalized joint variable. Depending on the scenario treated, $\ddot{r}_{\mathrm{des}} \in \mathbb{R}$ is able to express a simulated translation or rotation. Consequently, the constant scalar $m_{r,\mathrm{adm}} \in \mathbb{R}$ may represent the perceived mass or the moment of inertia of the mobile unit with respect to the vertical axis.

Likewise, $\tau_r^{\mathrm{ext}} \in \mathbb{R}$ and $\tau_r \in \mathbb{R}$ may refer to forces or torques acting on the mobile unit. Accordingly, τ_r^{ext} represents external joint torques, while τ_r expresses the aforementioned force-torque interface. Hence, it represents the connection between the mobile unit and the attached LWR.

Incorporating the admittance representing the basis in the overall EOM of the entire robotic system (refer to (4.42)), yields

$$\begin{bmatrix} m_{r,\mathrm{adm}} & \mathbf{0} \\ \boldsymbol{m}_{r,A}(\boldsymbol{y})^T & \boldsymbol{M}_A(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} \ddot{r}_{\mathrm{des}} \\ \ddot{\boldsymbol{q}} \end{bmatrix} + \begin{bmatrix} 0 & \mathbf{0} \\ \boldsymbol{c}_{r,A}(\boldsymbol{y},\dot{\boldsymbol{y}})^T & \boldsymbol{C}_A(\boldsymbol{q},\dot{\boldsymbol{q}}) \end{bmatrix} \begin{bmatrix} \dot{r}_{\mathrm{des}} \\ \dot{\boldsymbol{q}} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \boldsymbol{g}_q(\boldsymbol{y}) \end{bmatrix}}_{\boldsymbol{g}(\boldsymbol{y})} = \underbrace{\begin{bmatrix} \tau_r^{\mathrm{ext}} \\ \boldsymbol{\tau}_q^{\mathrm{ext}} \end{bmatrix}}_{\boldsymbol{\tau}^{\mathrm{ext}}} + \underbrace{\begin{bmatrix} \tau_r \\ \boldsymbol{\tau}_q \end{bmatrix}}_{\boldsymbol{\tau}} . \tag{5.3}$$

Let $\boldsymbol{\tau}$ represent the control input torques applied on both: the mobile unit and the LWR.

As [5] points out, the inertia and centrifugal coupling terms, $\boldsymbol{m}_{r,A}(\boldsymbol{y})^T \in \mathbb{R}^{7\times 1}$ and $\boldsymbol{c}_{r,A}(\boldsymbol{y},\dot{\boldsymbol{y}})^T \in \mathbb{R}^{7\times 1}$ can cause instability. Furthermore, the inertia coupling causes the system's inertia matrix to not be positive definite anymore. This motivates the introduction of a torque $\boldsymbol{\tau}^{\mathrm{comp}}$, which serves the purpose of eliminating the aforementioned coupling terms. Thus, the inertia matrix is guaranteed to be positive definite for the joint torque

$$\boldsymbol{\tau} = \underbrace{\begin{bmatrix} \tau_r^{\mathrm{hier}} \\ \boldsymbol{\tau}_q^{\mathrm{hier}} \end{bmatrix}}_{\boldsymbol{\tau}^{\mathrm{hier}}} + \underbrace{\begin{bmatrix} 0 \\ \boldsymbol{m}_{r,A}(\boldsymbol{y})^T\ddot{r}_{\mathrm{des}} + \boldsymbol{c}_{r,A}(\boldsymbol{y},\dot{\boldsymbol{y}})^T\dot{r}_{\mathrm{des}} \end{bmatrix}}_{\boldsymbol{\tau}^{\mathrm{comp}}} \in \mathbb{R}^8 \quad . \tag{5.4}$$

The other element making up the torque, namely $\boldsymbol{\tau}^{\mathrm{hier}}$, represents the joint torques resulting from the hierarchical impedance-based tracking control, which is defined in the next section.

Given the importance of this fact, the explanation given in Subsection 3.3.2 is recalled: the dynamics expressed by (5.2) do not represent a real system. The value chosen for

$m_{r,\text{adm}}$ represents the desired mass/inertia to be perceived by the system. Likewise, the desired acceleration $\ddot{r}_{\text{des}}$ is also only the product of a simulated system. In order to transmit the desired kinematic value to the real system, an additional velocity controller has to be implemented for the mobile unit. This velocity controller outputs the required joint torques for the actuating motors, which, in turn, enable the real system to behave according to the desired model.

Adopting the assumption of a high-gain velocity controller expressed in [7], the desired acceleration can be assumed to be realized immediately in the real physical system. Therefore, the assumption $\ddot{r} \approx \ddot{r}_{\text{des}}$ is made.

Consequently, the EOM are reduced to

$$\boldsymbol{M}(\boldsymbol{y})\ddot{\boldsymbol{y}} + \boldsymbol{C}(\boldsymbol{y}, \dot{\boldsymbol{y}})\dot{\boldsymbol{y}} + \boldsymbol{g}(\boldsymbol{y}) = \boldsymbol{\tau}^{\text{ext}} + \boldsymbol{\tau}^{\text{hier}} \quad , \tag{5.5}$$

with

$$\boldsymbol{M}(\boldsymbol{y}) = \begin{bmatrix} m_{r,\text{adm}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_A(\boldsymbol{y}) \end{bmatrix} \in \mathbb{R}^{8\times 8} \text{ and} \tag{5.6}$$

$$\boldsymbol{C}(\boldsymbol{y}, \dot{\boldsymbol{y}}) = \begin{bmatrix} 0 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_A(\boldsymbol{y}, \dot{\boldsymbol{y}}) \end{bmatrix} \in \mathbb{R}^{8\times 8} \quad . \tag{5.7}$$

Since the inertia's block entry $\boldsymbol{M}_A(\boldsymbol{y})$ is positive definite, the entire inertia matrix $\boldsymbol{M}(\boldsymbol{y})$ is guaranteed to be positive definite for positive value for $m_{r,\text{adm}}$ [60].

It shall also be noted, that the only block entry of the inertia matrix $\boldsymbol{M}(\boldsymbol{y})$ to be configuration dependent, is the lower right element $\boldsymbol{M}_A(\boldsymbol{y})$, representing the LWR's inertia. Further, the Coriolis matrix's $\boldsymbol{C}(\boldsymbol{y}, \dot{\boldsymbol{y}})$ only non-zero element, is likewise the lower right term $\boldsymbol{C}_A(\boldsymbol{y}, \dot{\boldsymbol{y}})$ representing the LWR. Therefore, despite the presence of a virtual mass $m_{r,\text{adm}}$ building the inertia matrix's $\boldsymbol{M}(\boldsymbol{y})$ derivative with respect to time, still fulfills the property expressed by (3.14). This, however, could only be the case for an admittance damping value of zero. This property's relevance becomes evident in the following section.

5.2. Hierarchical Impedance-Based Tracking Control

Throughout this section the strict hierarchical impedance-based tracking control implemented in the SwarmRail system is treated. On that note, it is resorted to the theoretical fundamentals provided in Subsection 3.3.1 and Section 3.4. Further, these concepts are extended by the steps laid down by Dietrich et al. [6].

What is more, the name of this controller is composed of three parts, as pointed out in the following. The hierarchical aspect of this controller has already been discussed in Section 3.4. Moreover, the term impedance refers to the fundamentals of the compliance controller discussed in Subsection 3.3.1. That is implying the implementation of a virtual spring-damper on the system while preserving the natural inertia [39]. The term "tracking" extends the concept introduced with the example in Subsection 3.3.1.

The aforementioned section aimed to achieve a desired configuration of the robot, by the means of impedance-based control. A tracking-based controller on the other hand, enables the robot to follow a desired trajectory [8]. In the following, the combination of these three ideas is treated.

5.2.1. Fundamentals

Given is a set of r tasks to be accomplished. The corresponding operational space velocities are given by

$$\dot{\boldsymbol{x}}_i = \boldsymbol{J}_i(\boldsymbol{y})\dot{\boldsymbol{y}} \in \mathbb{R}^{m_i} \quad , \tag{5.8}$$

for $1 \leq i \leq r$. The values m_i denote the dimension of the corresponding ith task.

As laid out in Section 3.4, the augmented Jacobian matrix

$$\boldsymbol{J}_i^{\text{aug}}(\boldsymbol{y}) = \begin{bmatrix} \boldsymbol{J}_1(\boldsymbol{y}) \\ \boldsymbol{J}_2(\boldsymbol{y}) \\ \vdots \\ \boldsymbol{J}_i(\boldsymbol{y}) \end{bmatrix} \quad \text{is used.} \tag{5.9}$$

At this stage we make two assumptions about the system:

Assumption 10: It is possible for the robotic system to execute all r tasks simultaneously. Hence,

$$\sum_{i=1}^{r} m_i = n \text{ holds.} \tag{5.10}$$

Assumption 11: The workspace does not contain any singularities.

This assumption is valid, given that trajectories can be planned to not contain any singularity. Consequently, singularity avoidance measures inside the controller are not required [6].

The augmented Jacobian $\boldsymbol{J}_r^{\text{aug}}(\boldsymbol{y}) \in \mathbb{R}^{n \times n}$ incorporating all defined tasks, possesses a particularly relevant property: due to Assumption 10, $\boldsymbol{J}_r^{\text{aug}}(\boldsymbol{y})$ is square. Additionally, the matrix is not singular, based on Assumption 11. Therefore, combining both aforementioned properties, it can thus be derived that $\boldsymbol{J}_r^{\text{aug}}(\boldsymbol{y})$ is invertible.

The dynamically consistent null space projectors, which induce the aforementioned hierarchy, are defined as

$$\boldsymbol{N}_i^{\text{aug}}(\boldsymbol{y}) = \begin{cases} \boldsymbol{I} & \text{for } i = 1 \\ \boldsymbol{I} - \boldsymbol{J}_{i-1}^{\text{aug}}(\boldsymbol{y})^T (\boldsymbol{J}_{i-1}^{\text{aug}}(\boldsymbol{y})^{M+})^T & \text{for } i = 2 \dots r \end{cases} \quad . \tag{5.11}$$

For this section a set of new augmented variables is introduced, such as the augmented task space velocities and accelerations, which are defined as

$$\dot{\boldsymbol{x}}_i^{\text{aug}} = \begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \\ \vdots \\ \dot{\boldsymbol{x}}_i \end{bmatrix}, \quad \ddot{\boldsymbol{x}}_i^{\text{aug}} = \begin{bmatrix} \ddot{\boldsymbol{x}}_1 \\ \ddot{\boldsymbol{x}}_2 \\ \vdots \\ \ddot{\boldsymbol{x}}_i \end{bmatrix} \quad . \tag{5.12}$$

What is more, (3.8) is hence generalized for the augmented operational space variables by

$$\dot{\boldsymbol{x}}_i^{\text{aug}} = \boldsymbol{J}_i^{\text{aug}}(\boldsymbol{y})\dot{\boldsymbol{y}} \quad . \tag{5.13}$$

Analogously, the mapping of wrenches expressed in the operational space to joint torques (refer to (3.15)) can be modified for augmented variables. For $\boldsymbol{F}_{\dot{\boldsymbol{x}}_i}^{\text{ext}}$ denoting the external wrenches acting on the operational space coordinate $\boldsymbol{x}_i$, the modified mapping is given by

$$\boldsymbol{\tau}^{\text{ext}} = \boldsymbol{J}_r^{\text{aug}}(\boldsymbol{y})^T \begin{bmatrix} \boldsymbol{F}_{\dot{\boldsymbol{x}}_1}^{\text{ext}} \\ \boldsymbol{F}_{\dot{\boldsymbol{x}}_2}^{\text{ext}} \\ \vdots \\ \boldsymbol{F}_{\dot{\boldsymbol{x}}_r}^{\text{ext}} \end{bmatrix} \quad . \tag{5.14}$$

Since this thesis aims to implement a tracking controller, all operational space coordinates $\boldsymbol{x}_i$ possess a time dependent desired value $\boldsymbol{x}_{i,\text{des}}(t)$. Consequently, the error between the current and desired configuration for a task i at time t is given by

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i(\boldsymbol{y}) - \boldsymbol{x}_{i,\text{des}}(t) \in \mathbb{R}^{m_i} \quad . \tag{5.15}$$

Analogously to (5.12), our desired velocities and accelerations can also be expressed in an augmented manner as

$$\dot{\boldsymbol{x}}_{i,\text{des}}^{\text{aug}} = \begin{bmatrix} \dot{\boldsymbol{x}}_{1,\text{des}} \\ \dot{\boldsymbol{x}}_{2,\text{des}} \\ \vdots \\ \dot{\boldsymbol{x}}_{i,\text{des}} \end{bmatrix}, \quad \ddot{\boldsymbol{x}}_i^{\text{aug}} = \begin{bmatrix} \ddot{\boldsymbol{x}}_{1,\text{des}} \\ \ddot{\boldsymbol{x}}_{2,\text{des}} \\ \vdots \\ \ddot{\boldsymbol{x}}_{i,\text{des}} \end{bmatrix} \quad . \tag{5.16}$$

The equivalent holds for the errors for each task. Hence,

$$\dot{\tilde{\boldsymbol{x}}}_i^{\text{aug}} = \begin{bmatrix} \dot{\tilde{\boldsymbol{x}}}_1 \\ \dot{\tilde{\boldsymbol{x}}}_2 \\ \vdots \\ \dot{\tilde{\boldsymbol{x}}}_i \end{bmatrix}, \quad \ddot{\tilde{\boldsymbol{x}}}_i^{\text{aug}} = \begin{bmatrix} \ddot{\tilde{\boldsymbol{x}}}_1 \\ \ddot{\tilde{\boldsymbol{x}}}_2 \\ \vdots \\ \ddot{\tilde{\boldsymbol{x}}}_i \end{bmatrix} \quad \text{also holds.} \tag{5.17}$$

5.2.2. Hierarchically Decoupled Equations of Motion

The hierarchical controller treated in Section 3.4 acknowledges that the velocities in the operational space are coupled in a top-down manner. Expressed differently: a joint torque applied for a particular task might cause a disturbing motion in a task of lower priority. The system is, consequently, said to be coupled. A coordinate transformation can thus be used in order to investigate the tasks of different priority levels independently from one another. The resulting coordinates v_i are less intuitive as they do not reflect real measurable velocities of the physical system. However, their benefit lies in their ability of simplifying the investigation of the respective priority levels. For this, the hierarchically decoupled Jacobian matrices

$$\hat{\boldsymbol{J}}_i(\boldsymbol{y}) = \boldsymbol{J}_i(\boldsymbol{y})\boldsymbol{N}_i^{\text{aug}}(\boldsymbol{y})^T \quad \text{are introduced.} \tag{5.18}$$

Using this allows us to perform the coordinate transformation

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_r \end{bmatrix} = \underbrace{\begin{bmatrix} \hat{\boldsymbol{J}}_1(\boldsymbol{y}) \\ \hat{\boldsymbol{J}}_2(\boldsymbol{y}) \\ \vdots \\ \hat{\boldsymbol{J}}_r(\boldsymbol{y}) \end{bmatrix}}_{\hat{\boldsymbol{J}}(\boldsymbol{y})} \dot{\boldsymbol{y}} \quad , \tag{5.19}$$

with $\hat{\boldsymbol{J}}(\boldsymbol{y}) \in \mathbb{R}^{n \times n}$ being the stacked hierarchically decoupled and invertible Jacobian matrix.

The application of this coordinate transformation on the robotic system affected by the admittance (refer to (5.5)), is given in full detail in Section B.4. The achieved results are the new hierarchically decoupled EOM

$$\boldsymbol{\Lambda}(\boldsymbol{y})\dot{\boldsymbol{v}} + \boldsymbol{\mu}(\boldsymbol{y}, \dot{\boldsymbol{y}})\boldsymbol{v} = \hat{\boldsymbol{J}}(\boldsymbol{y})^{-T}\left(\boldsymbol{\tau}^{\text{ext}} + \boldsymbol{\tau}^{\text{hier}} - \boldsymbol{g}(\boldsymbol{y})\right) \quad . \tag{5.20}$$

What is more, the decoupled inertia matrix is defined as

$$\begin{aligned} \boldsymbol{\Lambda}(\boldsymbol{y}) &= \hat{\boldsymbol{J}}(\boldsymbol{y})^{-T}\boldsymbol{M}(\boldsymbol{y})\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1} \in \mathbb{R}^{n \times n} \\ &= \text{diag}(\boldsymbol{\Lambda}_1(\boldsymbol{y}), \boldsymbol{\Lambda}_2(\boldsymbol{y}), \ldots, \boldsymbol{\Lambda}_r(\boldsymbol{y})) \quad . \end{aligned} \tag{5.21}$$

Each element $\boldsymbol{\Lambda}_i(\boldsymbol{y}) \in \mathbb{R}^{m_i \times m_i}$ of the block-diagonal matrix $\boldsymbol{\Lambda}(\boldsymbol{y})$ represents the positive definite and symmetric inertia matrix for the ith task.

The new transformed Coriolis matrix is then given by

$$\boldsymbol{\mu}(\boldsymbol{y}, \dot{\boldsymbol{y}}) = \hat{\boldsymbol{J}}(\boldsymbol{y})^{-T}\left(\boldsymbol{C}(\boldsymbol{y}, \dot{\boldsymbol{y}}) - \boldsymbol{M}(\boldsymbol{y})\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\dot{\hat{\boldsymbol{J}}}(\boldsymbol{y}, \dot{\boldsymbol{y}})\right)\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1} \in \mathbb{R}^{n \times n} \quad . \tag{5.22}$$

This matrix is fully occupied in general [6]. Therefore, it still presents a coupling between the individual dynamics of tasks in different priority levels. These are addressed by the control law, presented in the following sections.

5.2.3. From Hierarchically Decoupled to Operational Space Coordinates

The decoupled velocities v_i being unintuitive marks a major drawback such that a transformation of the EOM into the operational space is applied. Therefore, using the relations given by (5.19) and (5.13), the decoupled velocities can be expressed in dependence of the task space velocities as

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_r \end{bmatrix} = \underbrace{\hat{\boldsymbol{J}}(\boldsymbol{y}) \boldsymbol{J}_r^{\text{aug}}(\boldsymbol{y})^{-1}}_{\boldsymbol{B}(\boldsymbol{y})} \dot{\boldsymbol{x}}_r^{\text{aug}} \quad . \tag{5.23}$$

In this setting, the matrix $\boldsymbol{B}(\boldsymbol{y}) \in \mathbb{R}^{n \times n}$ is a block matrix, composed of the block elements $\boldsymbol{B}_{i,j}(\boldsymbol{y}) \in \mathbb{R}^{m_i \times m_j}$, respectively. Further, this matrix has a lower triangular shape. Moreover, another special property can be found in this matrix, namely, that its main diagonal is comprised of identity matrices. Therefore, the matrix $\boldsymbol{B}(\boldsymbol{y})$ represents the aforementioned top-down hierarchy.

It is important to note, that the identity matrices used, however, do not possess the same dimension. The identity matrix at the ith row and jth column, with $i = j$, is given by $\boldsymbol{I} \in \mathbb{R}^{m_i \times m_i}$. The shape of this inertia matrix allows two conclusions to be drawn:

1. Each velocity expressed by the hierarchically decoupled coordinates v_i contains its unweighted counterpart $\dot{\boldsymbol{x}}_i$ expressed in the operational space.
2. Decoupled velocities of lower priority contain weighted terms of operational velocities of higher prioritized tasks.

Combining these two insights also yields, that for the highest prioritized task, the velocities using operational or hierarchically decoupled coordinates are equal. Or expressed differently

$$v_i = \dot{\boldsymbol{x}}_i + \sum_{j=1}^{i-1} \boldsymbol{B}_{i,j}(\boldsymbol{y}) \dot{\boldsymbol{x}}_j \quad . \tag{5.24}$$

Deriving this equation with respect to time t yields

$$\dot{v}_i = \ddot{\boldsymbol{x}}_i + \sum_{j=1}^{i-1} \left(\boldsymbol{B}_{i,j}(\boldsymbol{y}) \ddot{\boldsymbol{x}}_j + \dot{\boldsymbol{B}}_{i,j}(\boldsymbol{y}, \dot{\boldsymbol{y}}) \dot{\boldsymbol{x}}_j \right) \quad . \tag{5.25}$$

Analogously, a transition from the external joint torques $\hat{\boldsymbol{J}}(\boldsymbol{y})^{-T} \boldsymbol{\tau}^{\text{ext}}$provided by (5.20), to the decoupled coordinates can be performed. Utilizing (5.14),

$$\begin{bmatrix} \boldsymbol{F}_{v_1}^{\text{ext}} \\ \boldsymbol{F}_{v_2}^{\text{ext}} \\ \vdots \\ \boldsymbol{F}_{v_r}^{\text{ext}} \end{bmatrix} = \hat{\boldsymbol{J}}(\boldsymbol{y})^{-T} \boldsymbol{\tau}^{\text{ext}} = \underbrace{\hat{\boldsymbol{J}}(\boldsymbol{y})^{-T} \boldsymbol{J}_r^{\text{aug}}(\boldsymbol{y})^{T}}_{\boldsymbol{E}(\boldsymbol{y})} \begin{bmatrix} \boldsymbol{F}_{\dot{x}_1}^{\text{ext}} \\ \boldsymbol{F}_{\dot{x}_2}^{\text{ext}} \\ \vdots \\ \boldsymbol{F}_{\dot{x}_r}^{\text{ext}} \end{bmatrix} \tag{5.26}$$

can be derived, with

$$\boldsymbol{E}(\boldsymbol{y}) = \boldsymbol{B}(\boldsymbol{y})^{-T} \quad . \tag{5.27}$$

Consequently, the matrix $\boldsymbol{E}(\boldsymbol{y})$ possesses an upper triangular structure, with the diagonal submatrices consisting of identity matrices. Therefore, the cross couplings are reversed. The ith wrench in decoupled coordinates can be expresses similarly to (5.24), namely as

$$\boldsymbol{F}_{v_i}^{\text{ext}} = \boldsymbol{F}_{\dot{x}_i}^{\text{ext}} + \sum_{j=i+1}^{r} \boldsymbol{E}_{i,j}(\boldsymbol{y}) \boldsymbol{F}_{\dot{x}_j}^{\text{ext}} \quad . \tag{5.28}$$

5.2.4. Control Law

The control law as proposed by Dietrich et al. [6] is comprised of three different elements

$$\boldsymbol{\tau}^{\text{hier}} = \boldsymbol{g}(\boldsymbol{y}) + \boldsymbol{\tau}_{\mu} + \sum_{i=1}^{r} \boldsymbol{N}_i^{\text{aug}} \boldsymbol{J}_i^T \boldsymbol{F}_i^{\text{ctrl}} \quad . \tag{5.29}$$

The first term, namely $\boldsymbol{g}(\boldsymbol{y}) \in \mathbb{R}^n$, serves as a compensation for all effects attributed to gravity. The second term, addresses the information provided in Subsection 5.2.2: the Coriolis matrix $\boldsymbol{\mu}(\boldsymbol{y}, \dot{\boldsymbol{y}})$ still presents a coupling between the dynamics of tasks in different priority levels. In order to prevent these couplings, the term $\boldsymbol{\tau}_{\mu} \in \mathbb{R}^n$ is introduced. It is defined as

$$\boldsymbol{\tau}_{\mu} = \sum_{i=1}^{r} \left(\hat{\boldsymbol{J}}_i(\boldsymbol{y})^T \left(\sum_{j=1}^{i-1} \boldsymbol{\mu}_{i,j} \boldsymbol{v}_j + \sum_{j=i+1}^{r} \boldsymbol{\mu}_{i,j} \boldsymbol{v}_j \right) \right) \quad . \tag{5.30}$$

The last term of the control law incorporates the wrenches in the operational space which are needed for the execution of all r tasks. Implementing (5.29) inside (5.20) yields

$$\boldsymbol{\Lambda}_i(\boldsymbol{y}) \dot{\boldsymbol{v}}_i + \boldsymbol{\mu}_{i,i}(\boldsymbol{y}, \dot{\boldsymbol{y}}) \boldsymbol{v}_i = \boldsymbol{F}_i^{\text{ctrl}} + \boldsymbol{F}_{v_i}^{\text{ext}} \quad , \tag{5.31}$$

for $i = 1 \dots r$. Alternatively, the EOM can be expressed inside the operational space as

$$\boldsymbol{\Lambda}_i(\boldsymbol{y}) \ddot{\boldsymbol{x}}_i + \boldsymbol{\mu}_{i,i}(\boldsymbol{y}, \dot{\boldsymbol{y}}) \dot{\boldsymbol{x}}_i + \gamma_i(\boldsymbol{y}, \dot{\boldsymbol{y}}) \begin{bmatrix} \dot{\boldsymbol{x}}_{i-1}^{\text{aug}} \\ \ddot{\boldsymbol{x}}_{i-1}^{\text{aug}} \end{bmatrix} = \boldsymbol{F}_i^{\text{ctrl}} + \boldsymbol{F}_{v_i}^{\text{ext}} \quad , \tag{5.32}$$

with

$$\gamma_i(\boldsymbol{y}, \dot{\boldsymbol{y}}) = (\boldsymbol{\Gamma}_{i,1}, \dots, \boldsymbol{\Gamma}_{i,i-1}, \boldsymbol{\Psi}_{i,1}, \dots, \boldsymbol{\Psi}_{i,i-1}) \tag{5.33}$$

$$\boldsymbol{\Gamma}_{i,j}(\boldsymbol{y}, \dot{\boldsymbol{y}}) = \boldsymbol{\mu}_{i,i} \boldsymbol{B}_{i,j} + \boldsymbol{\Lambda}_i \dot{\boldsymbol{B}}_{i,j} \tag{5.34}$$

$$\boldsymbol{\Psi}_{i,j}(\boldsymbol{y}) = \boldsymbol{\Lambda}_i \boldsymbol{B}_{i,j} \quad . \tag{5.35}$$

The derivation of the new EOM inside the operational space is provided in Section B.5.

The individual control forces are chosen accordingly to [6] to be

$$\boldsymbol{F}_i^{\text{ctrl}} = \boldsymbol{\Lambda}_i \ddot{\boldsymbol{x}}_{i,\text{des}} + \boldsymbol{\mu}_{i,i} \dot{\boldsymbol{x}}_{i,\text{des}} - \boldsymbol{D}_i \dot{\tilde{\boldsymbol{x}}}_i - \boldsymbol{K}_i \tilde{\boldsymbol{x}}_i + \boldsymbol{\gamma}_i(\boldsymbol{y}, \dot{\boldsymbol{y}}) \begin{bmatrix} \dot{\boldsymbol{x}}_{i-1,\text{des}}^{\text{aug}} \\ \ddot{\boldsymbol{x}}_{i-1,\text{des}}^{\text{aug}} \end{bmatrix} - \boldsymbol{F}_{i,\text{ext}}^{\text{ctrl}} \quad . \tag{5.36}$$

With the stiffness matrix $\boldsymbol{K}_i \in \mathbb{R}^{m_i \times m_i}$ and the damping matrix $\boldsymbol{D}_i \in \mathbb{R}^{m_i \times m_i}$ for the impedance-based control being chosen symmetric and positive definite.

The last term, $\boldsymbol{F}_{i,\text{ext}}^{\text{ctrl}}$, is the optional annihilation of external wrench couplings, as described in (5.28). Therefore, the last term can be calculated in twofold forms, namely

$$\boldsymbol{F}_{i,\text{ext}}^{\text{ctrl}} = \begin{cases} \boldsymbol{0} & \text{for case 1} \\ \sum_{j=i+1}^{r} \boldsymbol{E}_{i,j}(\boldsymbol{y}) \boldsymbol{F}_{\dot{\boldsymbol{x}}_j}^{\text{ext}} & \text{for case 2} \end{cases} , \tag{5.37}$$

depending on whether or not it is chosen to either not compensate (case 1) or compensate the external wrench couplings (case 2).

Applying $\boldsymbol{F}_i^{\text{ctrl}}$ on the EOM expressed by (5.32) yields

$$\boldsymbol{\Lambda}_i(\boldsymbol{y}) \ddot{\tilde{\boldsymbol{x}}}_i + \left(\boldsymbol{\mu}_{i,i}(\boldsymbol{y}, \dot{\boldsymbol{y}}) + \boldsymbol{D}_i \right) \dot{\tilde{\boldsymbol{x}}}_i + \boldsymbol{K}_i \tilde{\boldsymbol{x}}_i = -\boldsymbol{\gamma}_i(\boldsymbol{y}, \dot{\boldsymbol{y}}) \begin{bmatrix} \dot{\tilde{\boldsymbol{x}}}_{i-1}^{\text{aug}} \\ \ddot{\tilde{\boldsymbol{x}}}_{i-1}^{\text{aug}} \end{bmatrix} + \boldsymbol{F}_{v_i}^{\text{ext}} \tag{5.38}$$

for case 1 and

$$\boldsymbol{\Lambda}_i(\boldsymbol{y}) \ddot{\tilde{\boldsymbol{x}}}_i + \left(\boldsymbol{\mu}_{i,i}(\boldsymbol{y}, \dot{\boldsymbol{y}}) + \boldsymbol{D}_i \right) \dot{\tilde{\boldsymbol{x}}}_i + \boldsymbol{K}_i \tilde{\boldsymbol{x}}_i = -\boldsymbol{\gamma}_i(\boldsymbol{y}, \dot{\boldsymbol{y}}) \begin{bmatrix} \dot{\tilde{\boldsymbol{x}}}_{i-1}^{\text{aug}} \\ \ddot{\tilde{\boldsymbol{x}}}_{i-1}^{\text{aug}} \end{bmatrix} + \boldsymbol{F}_{\dot{\boldsymbol{x}}_i}^{\text{ext}} \tag{5.39}$$

for case 2. For the scope of this thesis, both cases are covered and compared with regards to their performance.

For case 1, this system has been proven to be asymptotically stable with regard to its equilibrium, i.e. for the case of all errors inside the operational space being zero. For case 2 the asymptotic stability of the equilibrium can also be proven for the case of constant external forces. The interested reader is advised to refer to [6] for the respective proofs of the aforementioned theorems. What is more, a special focus is set on the matter of fact, that the property (3.14) serves as a key requirement for the aforementioned proof. Hence, the latter is only applicable due to the omission of the admittance damping value $d_{r,\text{adm}}$ (refer to the explanation provided in Section 5.1).

Despite this benefit, case 2 suffers from the requirement of precise knowledge of the external wrenches. Therefore, additional sensitive force-torque sensors are required. One proposal of deriving the external wrenches based on the sensor readings, is given by the estimation according to Iskandar et al. [66].

It shall also be noted that, despite the external force coupling being introduced for case 2, top down disturbances are still possible. This can be explained by the factor $\boldsymbol{\gamma}_i(\boldsymbol{y}, \dot{\boldsymbol{y}})$ in (5.39) which introduces the tracking errors of higher prioritized tasks into

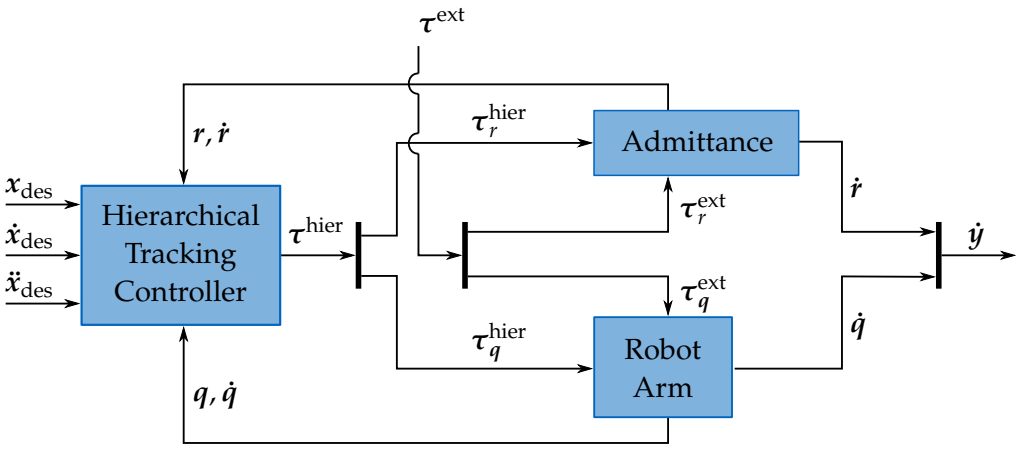


Figure 5.1.: Structure of the implemented controller used in the scope of the physical simulation.

the EOM of a lower prioritized task. Consequently, any non-constant external force $\boldsymbol{F}_{\dot{x}_k}^{ext}$ which induces an acceleration $\ddot{\boldsymbol{x}}_k$ can introduce disturbances to all tasks of lower priority [6, 19]. This issue is addressed by Wu et al. [19], but was not considered in the scope of this thesis.

The overall structure of the implemented controller can be seen in Figure 5.1. Note that for the application of the controller inside the real physical system the admittance would only produce simulated desired velocities. These velocities in turn would be fed to a velocity controller, which would provide torques for the real mobile unit. However, given that we assumed the desired velocities to be equal to ones inside the real system, this part can be omitted. Additionally, the torques provided by the hierarchical tracking controller would be fed into a torque controller, which would apply the desired torques on the physical system.

6. Controller Validation

Having established the theoretical fundamentals of the chosen controller design, this chapter proceeds to test the implementation on tracking tasks simulated using MATLAB and Simulink [67, 68]. This chapter is divided into three sections: the first section covers the parameter selection underlying the modeling of the kinematic constraints, which were mentioned in Section 4.3. The section following that covers the tasks implemented in our simulations, also presenting our parameter choices for both, the RS and TS. The last section covers the TS in extensive detail, analyzing the controller's performance under the presence of disturbances. During this analysis, the effects of adding a term compensating external force couplings into the introduced control law are examined.

The attentive reader might have noticed the absence of a detailed coverage of the RS in the last part of this chapter. However as explained earlier, the main challenge for the SwarmRail system is encountered whenever the mobile unit traverses a crossing point and is exposed to both, sudden position changes and impact forces. As these conditions are not present for the RS, covering this scenario in detail would not provide significant new insights to the papers [5–7] that served as reference for the development of the controller. Moreover, as all sides possess equal lengths and sensor equipment, a change of orientation for the rectangular grid is irrelevant inside a rectangular rail-network. Since the SwarmRail system proposes an omni-directional mobile unit, the robot can progress in any direction without the need of rotation. For completion sake, however, the simulation results for the RS can be found in the Appendix (refer to Sections C.1 and C.2).

6.1. Parameters Kinematic Constraints

In this section the parameter choices for the model representing the kinematic constraints (refer to Section 4.3) are presented. Regarding the singular perturbation approach, the parameter $\epsilon = 0.001$ is chosen. The stiffness parameters k_i, for $i = 1, 2$ (refer to (4.6)), are dependent on which type of motion is being "blocked". For restricted translational motions, k_i is chosen to be equal to $100\frac{\mathrm{N}}{\mathrm{m}}$. For "blocked" rotational motions, however, the stiffness is chosen to be equal to $1\frac{\mathrm{Nm}}{\mathrm{rad}}$. The damping parameter is chosen accordingly to (4.7).

The vertical spring-damper is implemented with a stiffness of $5000000\frac{\mathrm{N}}{\mathrm{m}}$ and a damping parameter, which is also chosen according to (4.7), for a damping ratio of 1. With these choices of a large stiffness and a damping ratio resulting in a critically damped system, we aim to enforce the robotic system to be kept at a constant altitude. Addi-

tionally, after being subjected to a disturbance, the spring-damper enforces the robot to return to the equilibrium as fast as possible without oscillations (refer to the explanation given in Section 4.3). As explained in Subsection 4.4.2, the altitude drop is modeled as a temporary removal of the vertical spring-damper for the length of a falling time t_{Fall}. For our experiments, this time is set to 0.03s. This equates to an altitude drop of 0.5cm in total, occurring for each passing of a crossing point.

6.2. Analysed Tasks

The controller validation is performed for a set of four hierarchically ordered tasks. For both, the RS and TS, the two highest prioritized tasks cover the TCP's translational and rotational motion control, respectively. For both scenarios, the third level task is concerned with the mobile unit's motion. Due to the mobile unit's differing motion capabilities, however, a distinction has to be made for each individual scenario. For the RS, the third task covers the mobile unit's rotation about the vertical z-axis. The TS, on the other hand, handles the translation of the mobile unit. For the scope of this thesis, analogously to the explanation of the kinematic constraints provided in Section 4.3, the only translation observed is along the x-direction of the world-frame (refer to Figure 4.1). Lastly, controlling the first joint's position of the attached LWR is set to be the lowest prioritized task. The task hierarchy, along with the corresponding dimension and selected gains for both tasks, are summarized in Table 6.1.

Table 6.1.: Task descriptions and gains used for the simulation. The lower the level, the higher the prioritization of the task.

Lvl[1]	Dim[2]	Task Description	Stiffness	Damping Ratio
1	3	TCP translational motion	$\text{diag}(4500, 4500, 4500)\frac{\text{N}}{\text{m}}$	0.9
2	3	TCP rotational motion	$\text{diag}(800, 800, 800)\frac{\text{Nm}}{\text{rad}}$	0.9
3	1	Mobile unit motion	$4500\frac{\text{Nm}}{\text{rad}} / \frac{\text{N}}{\text{m}}$	0.9
4	1	First joint LWR position	$300\frac{\text{Nm}}{\text{rad}}$	0.7

The damping matrices of the impedance controller are obtained by making use of the Double Diagonalization approach, as proposed by Albu-Schäffer [69], using the damping ratios provided in Table 6.1. At this stage it should be emphasized, that the operational space dimensions for all tasks combined is equal to eight. Therefore, the summed up dimensions of all tasks are equal to the number of DOF of the robotic system. Therefore, this task selection fulfills Assumption 10.

Additionally, the admittance is chosen to model the mobile unit as a cube with a virtual mass of $m_{\text{vir}} = 15\text{kg}$, a length of $l = 0.362\text{m}$ and a width of $w = 0.362\text{m}$. For the RS, the moment of inertia about the vertical z-axis, is equal to

[1]Level
[2]Dimension

$$m_{r,\text{adm}} = \frac{m_{\text{vir}}}{12}\left(l^2 + w^2\right) = 0.328\text{kg} \cdot \text{m}^2 \quad . \tag{6.1}$$

For the TS the admittance mass is simply equal to the modeled virtual mass, namely $m_{r,\text{adm}} = 15\text{kg}$.

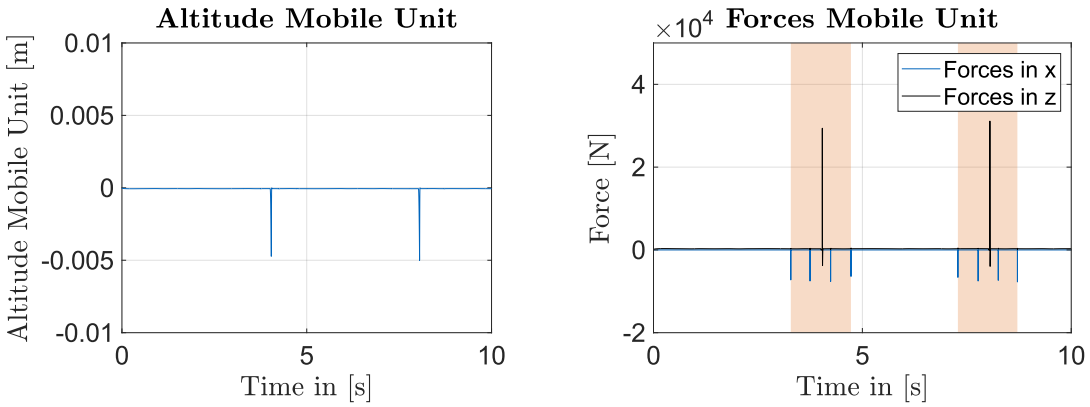
6.3. Simulations for the Translational Scenario

In this section the tracking capabilities of the SwarmRail robot, under the presence of sudden positional changes and high impact forces, are discussed. This is achieved by specifying the desired trajectory of Task 3 to be a translation of the mobile unit with a constant desired velocity in x-direction. Therefore, the robot moves across the rail-network passing the crossing points. We choose our velocity to be equal to the highest value possible, at which the demonstrator is still capable of reliably centering itself. By this choice, the resulting external impact forces acting on the system are modeled as high as possible for a correctly operating robot. Through this, the worst possible case is simulated, which in turn gives insights about the controllers performance under extreme conditions.

The original SwarmRail paper proposed a maximal translation speed of $0.5\frac{\text{m}}{\text{s}}$ inside the straight segments and $0.3\frac{\text{m}}{\text{s}}$ for the traversal of the crossing points, respectively [4, 38]. However, further experiments conducted by our group recommend even lower speeds for reliable performance. Having employed the mobile unit to different rail-networks under various conditions has shown, that the optical sensors used for the centering process are sensitive to varying lighting conditions. In order to compensate for this issue, the optimal translation speed at the crossing points was deemed most reliable for a maximal speed of $0.25\frac{\text{m}}{\text{s}}$. Therefore, the simulations set the mobile unit to drive along the rails in a straight line with the aforementioned constant speed of $0.25\frac{\text{m}}{\text{s}}$.

For our rail-network, characterized by a distance of 1m between consecutive crossing points, this velocity leads to the traversal of two crossing points for each simulation run lasting 10s. Based on the parameters introduced at the beginning of this chapter, the spring-dampers' impact on the system can be observed in Figure 6.1.

For most of the simulated time the mobile unit's altitude measured along the z-axis is kept at a range of [-0.069; -0.047]mm. The two exceptions, occurring at the four and eight second mark, can be attributed to the drop in altitude of 5mm, caused by the removal of the vertical spring-damper (refer to Figure 6.1a). The reapplication of said spring-damper is visualized by the black force peak at the center of the orange area in Figure 6.1b. This force acts along the vertical z-axis, therefore against the gravity. Consequently the mobile unit is pushed back to the original altitude. Note that after acting in the positive direction with a magnitude of 30kN, the mobile unit overshoots its equilibrium position. Therefore, the force in z transitions to a negative value with a magnitude of 3.8kN to adjust the robot's position. At the peak of the mentioned overshoot, the robot reaches a maximal altitude of -0.008mm, therefore it is still in close proximity to the equilibrium.



(a) Altitude changes resulting from the traversal of crossing points.

(b) Forces acting on the mobile unit resulting from the impact of the system with the consequent rails and the modeled vertical kinematic constraint.

Figure 6.1.: Influences on the mobile unit when traversing a crossing point.

The remaining four peaks surrounding the vertical force, denote the horizontal collision forces acting oppositely to the mobile unit's driving direction. For our example, this would be in negative x-direction. This force is incorporated in the vector of external forces acting on the mobile unit. The time frame in which the entire crossing point traversal takes place, is highlighted by an orange area. It spans the time from the first to the last frontal collision. In order to set the focus on the influence the passage of the crossing point has on the tracking performance, this shaded area is also used for the plots to follow.

At this point it should be highlighted, that the vertical force models a kinematic constraint keeping the robot from falling. It does not reflect a physical collision between the mobile unit with the rails, as this is covered by the forces in x. The peaks of the force in z should hence be interpreted as a possibility of achieving sudden position changes, instead of a physically originated external force occurring on the system. We assume the robot to be equipped with an IMU, as described by Görner et al. [4]. Therefore, the vertical acceleration can be assumed to be known. This enables for the derivation of the change in altitude of the mobile unit, allowing the robot to react to the sudden disturbance by an adjustment of its joint positions to counteract the effects of the altitude drop inside the performed tasks. Therefore, when this thesis refers to "external force compensation", only the coupling terms of the collision forces applied in x are compensated. Consequently, the modeled external force is directly applied at the task Level 3 and presents cross coupling with all tasks of higher priorities (refer to (5.37)).

The effects of the spring-damper systems constraining the mobile unit's translation in y-direction and rotation about the vertical axis z, are depicted in Figure 6.2. During the entire duration of the simulation, the mobile unit's deviation from the

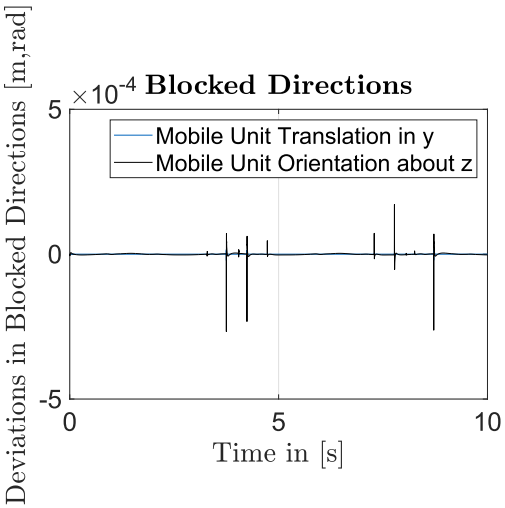


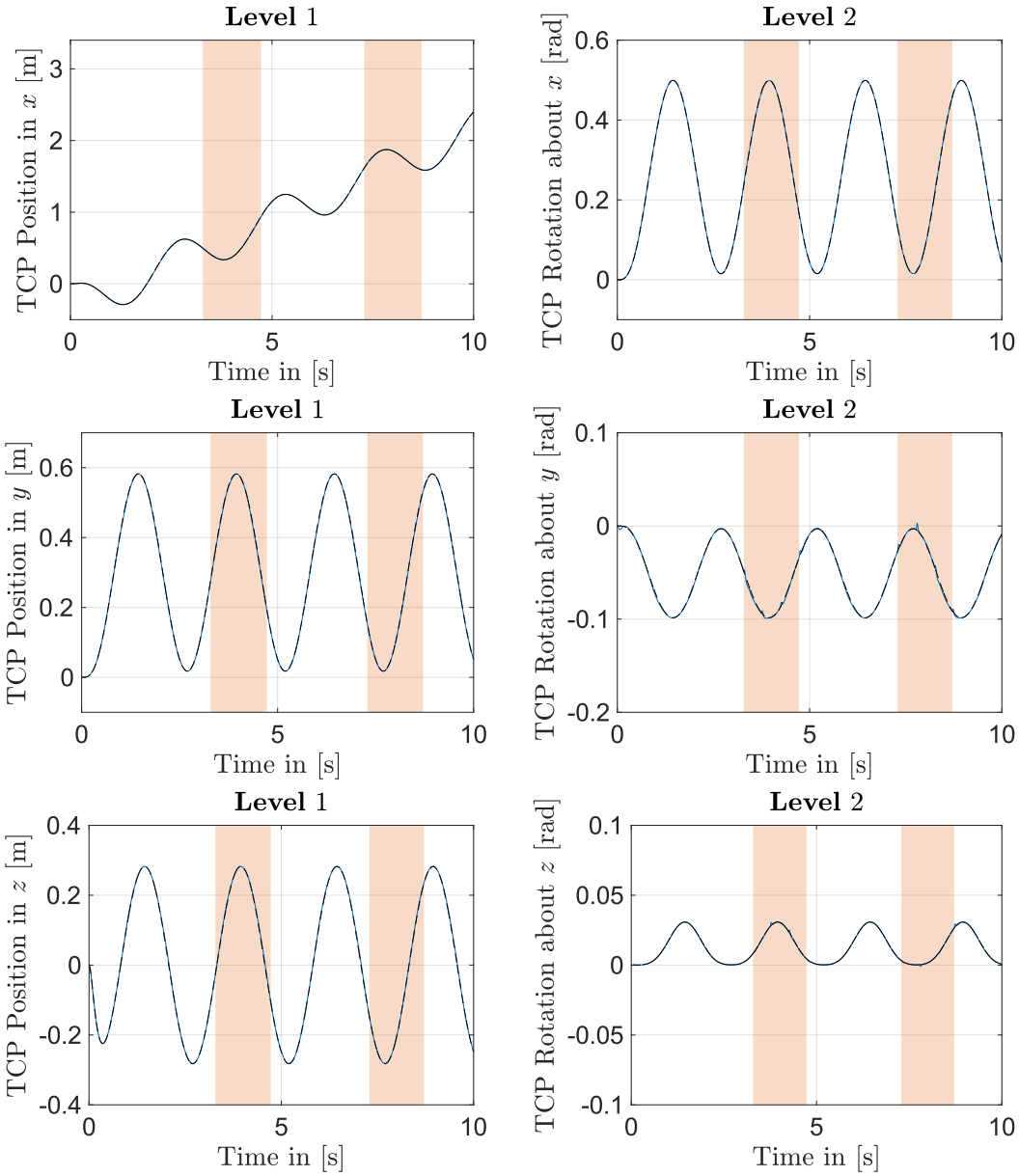
Figure 6.2.: The mobile unit's deviations in the directions of the constrained DOF.

equilibrium $(y_{u,0}, \varphi_{u,0}) = (0, 0)$ is kept small. For most of the simulation the deviations from said equilibrium remain inside a range of [-67; +81]nm for translation in y and [-0.0066; +0.0095]rad for the rotation angle about the z-axis. Exceptions are encountered within the time intervals in which crossing points are passed. During the latter, the maximal deviation in y is achieved at a magnitude of 0.021mm, while the maximal deviation of the rotation angle is given by a magnitude of $2.67 \cdot 10^{-4}$rad. Despite representing relative peaks in their respective own courses, they remain at least two orders of magnitude below the average tracking errors which are presented later in this section. Therefore, their influence on the observed errors is negligible. The assumption of neglecting these DOF as derived in Section 4.3 is therefore indeed justifiable.

Having treated the preliminary conditions and assumptions, this section progresses with the presentation of the desired trajectories. For the scope of this section, only the trajectories depicted in Figure 6.3 are examined. These trajectory are tracked twice: once by the controller without and once by the version with external force compensation being applied. The figures include the simulated as well as the desired trajectories for all four tasks. It is important to mention, that the trajectory of Level 2 is expressed by the three Euler angles representing elementary rotations about the three axes of the world frame. These are extracted from the rotation matrix representing the TCP's orientation, based on the work of Day [43].

It should be noted, that Figure 6.3 demonstrates the tracking performed by the controller without external force compensation only. Naturally, the compensating counterpart (described by case 2 in (5.37)) which serves as comparison is evaluated using the same desired trajectory. However, these plots did not reveal additional information for the scope of this thesis. Hence they were omitted in this section. The interested reader is advised to refer to the Appendix, namely Section C.3, for illustrative comprehension of bespoken case.

Figure 6.3 shows, that the controller is overall able to track the desired trajectories well, even without external force compensation. In order to obtain further more meaningful results, the underlying tracking errors are analyzed thoroughly. Firstly, we provide a description of the errors for the control variant without external force compensation. Building upon this, the tracking errors for the controller that uses external force compensation are introduced. For the latter, an evaluation as well as a comparative analysis with the uncompensated controller is performed.



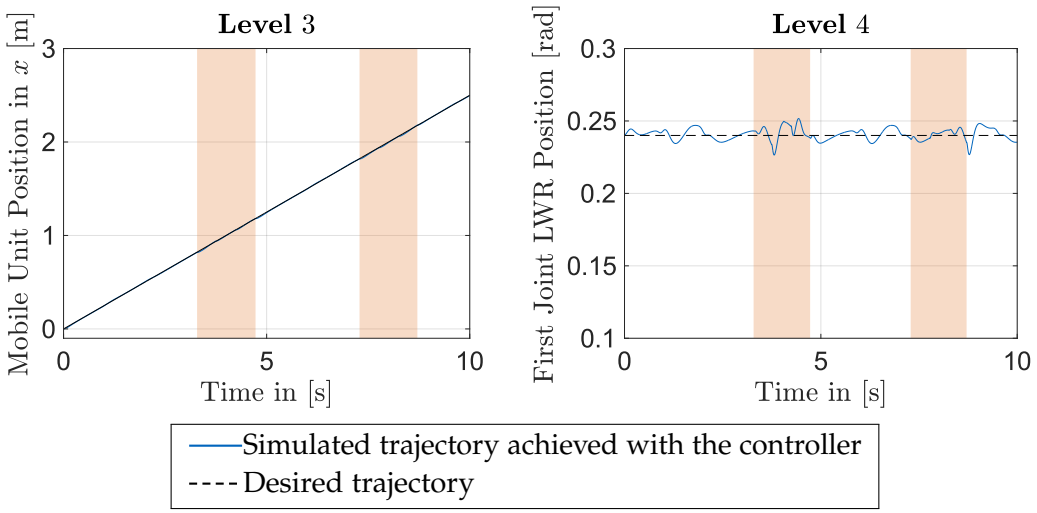


Figure 6.3.: Tracked trajectories for the controller without force compensation.

6.3.1. Case 1: Controller without Compensation

When analyzing the errors over time, the influence of the forces and altitude changes inside the crossing points become evident. For this analysis, the focus is initially set on the tasks of tracking the TCP-position in x (refer to Figure 6.4a). For the position in x, each crossing passage is accompanied by four negative peaks, representing large position changes. The effect of the mobile unit's altitude change in z is noticeable for the TCP in x, but does not severely affect the error. Instead, it is through the collision forces acting in the x-direction that the controller is disturbed the most. For the TCP position in the z-direction, however, the opposite holds: the peaks of greatest magnitude occur simultaneously to the drop of the mobile unit (refer to Figure 6.4e).

The aforementioned relationships are obvious: once the mobile unit is disturbed in a direction, this disturbance also acts in the same direction for the attached LWRs TCP located on top of the robot. Comparing the magnitude of these two correlated errors, however, provides important insights: while the collisions cause the mobile unit to retract by distances up to 10mm along the x-axis (refer to Figure 6.4i) the TCP, however, is only pushed back by a maximum of 4mm in x-direction. This equates to a reduction of 40% between the deviation peaks occurring at the mobile unit and the TCP. These observations can be transferred analogously to the deviation in z. Although the mobile unit experiences a sudden altitude change of 5mm in the negative z-direction, the TCP is only affected by a drop of 2.8mm with respect to the same axis. Therefore, the deviation is reduced to 56% of the mobile unit's deviation.

The aforementioned disturbances are also visible for the error in the TCP's orientation, as depicted in Figure 6.4g. During the passage of the crossing points, the orientation errors experience sudden and abrupt deviation peaks, aligned with the disturbances on the mobile unit. Most noticeably, among the three different axes, the rotation angle

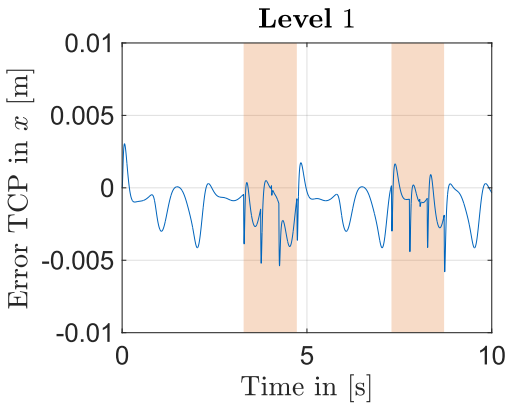
about the y-axis experiences the greatest disturbances, reaching peaks of about 0.0084rad for the times in which the collision forces act on the mobile unit.

6.3.2. Case 2: Controller with External Force Coupling Compensation

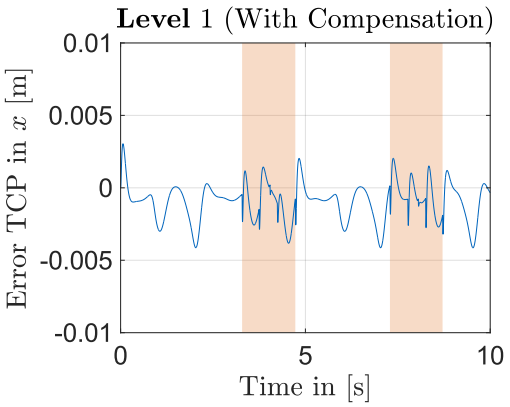
By introducing the external force compensation term inside the control law, the overall behavior observed during the entire translation at the straight segments does not present significant changes. That is to be expected, given that inside the straight segment no disturbing external forces act on the system. The absolute errors averaged over time for each level, depicted by Figure 6.5a for the translations and Figure 6.5b for the rotations, confirm this notion. The average absolute errors over the entire duration of the simulation barely change once external force compensation is added. For Level 1 the error is reduced by only 0.1mm in total for each direction. Level 2 only experiences an error reduction of 0.1rad for the angle representing the rotation about y, the other values remain unchanged. Likewise, the error in Level 3 remains the same and Level 4 possesses an average absolute error, which is higher by 0.2rad.

Inside the crossing points, however, these modifications show their effect. Therefore, the focus is once more directed back to the peaks caused by the mobile unit's passage of the crossing points. Particularly the errors depicted in both, Figure 6.4b and Figure 6.4h are to be highlighted. Before applying the compensation term, the collisions in x caused a change in position of up to 4mm at the TCP along the same x-axis. After applying the compensation, the maximal error peak is reduced to a magnitude of 2mm, equating for a reduction by the factor of 2. Compared to the deviation in the x-direction observable for the mobile unit, the TCP error only represents 20% of the mobile unit's error along the same axis. For the TCP's error regarding the rotation about the y-axis, the original striking peaks were reduced by 50%, compared with the first control variant analyzed. The errors for the TCP position in y and z (refer to Figure 6.4d and Figure 6.4f) also follow a smoother path, with the peak magnitudes caused by the collisions being reduced. Since these deviation were small even before compensation, this change is not as striking. Furthermore, the peaks at the center of the highlighted areas, representing the effects of the altitude change, have remained unchanged by the compensation term. This is most noticeable for the peak in the TCP's deviation in z, which still possesses a magnitude of 2.8mm (refer to Figure 6.4f).

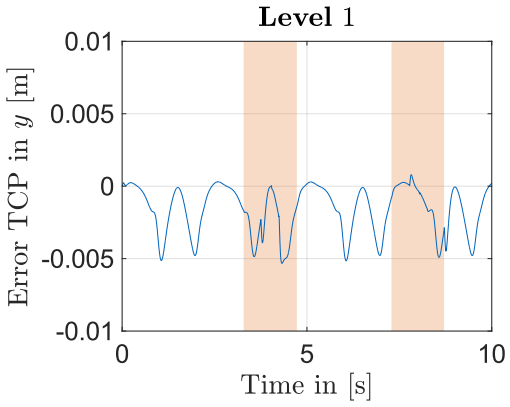
When inspecting the maximal absolute errors reached at each task, it is noticeable, that for the two highest prioritized tasks the errors were always able to be reduced with exception of the TCP's deviation in z (compare Figure 6.6a and Figure 6.6b). Meanwhile the error for Level 3 experiences a slight rise of 0.1mm and Level 4's error increases by $0.39 \cdot 10^{-2}$rad.


Level 1
Error TCP in x [m]
0.01
0.005
0
-0.005
-0.01
0
5
10
Time in [s]

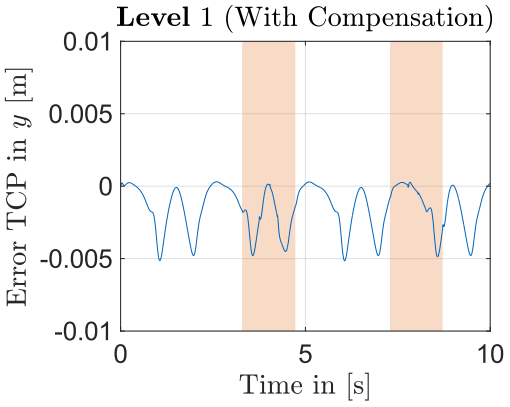
(a)


Level 1 (With Compensation)
Error TCP in x [m]
0.01
0.005
0
-0.005
-0.01
0
5
10
Time in [s]

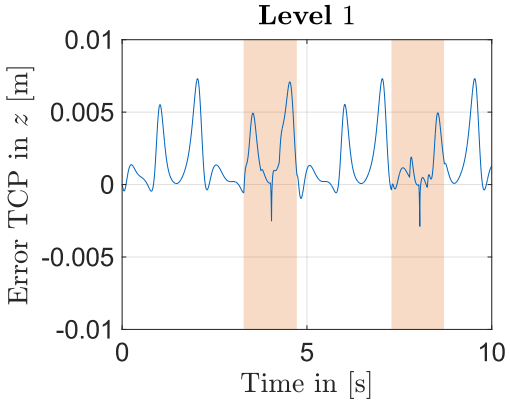
(b)


Level 1
Error TCP in y [m]
0.01
0.005
0
-0.005
-0.01
0
5
10
Time in [s]

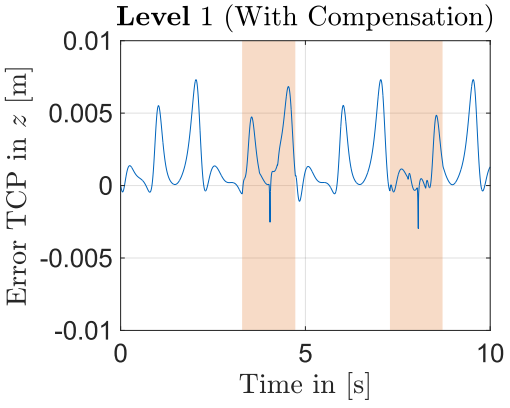
(c)


Level 1 (With Compensation)
Error TCP in y [m]
0.01
0.005
0
-0.005
-0.01
0
5
10
Time in [s]

(d)


Level 1
Error TCP in z [m]
0.01
0.005
0
-0.005
-0.01
0
5
10
Time in [s]

(e)


Level 1 (With Compensation)
Error TCP in z [m]
0.01
0.005
0
-0.005
-0.01
0
5
10
Time in [s]

(f)

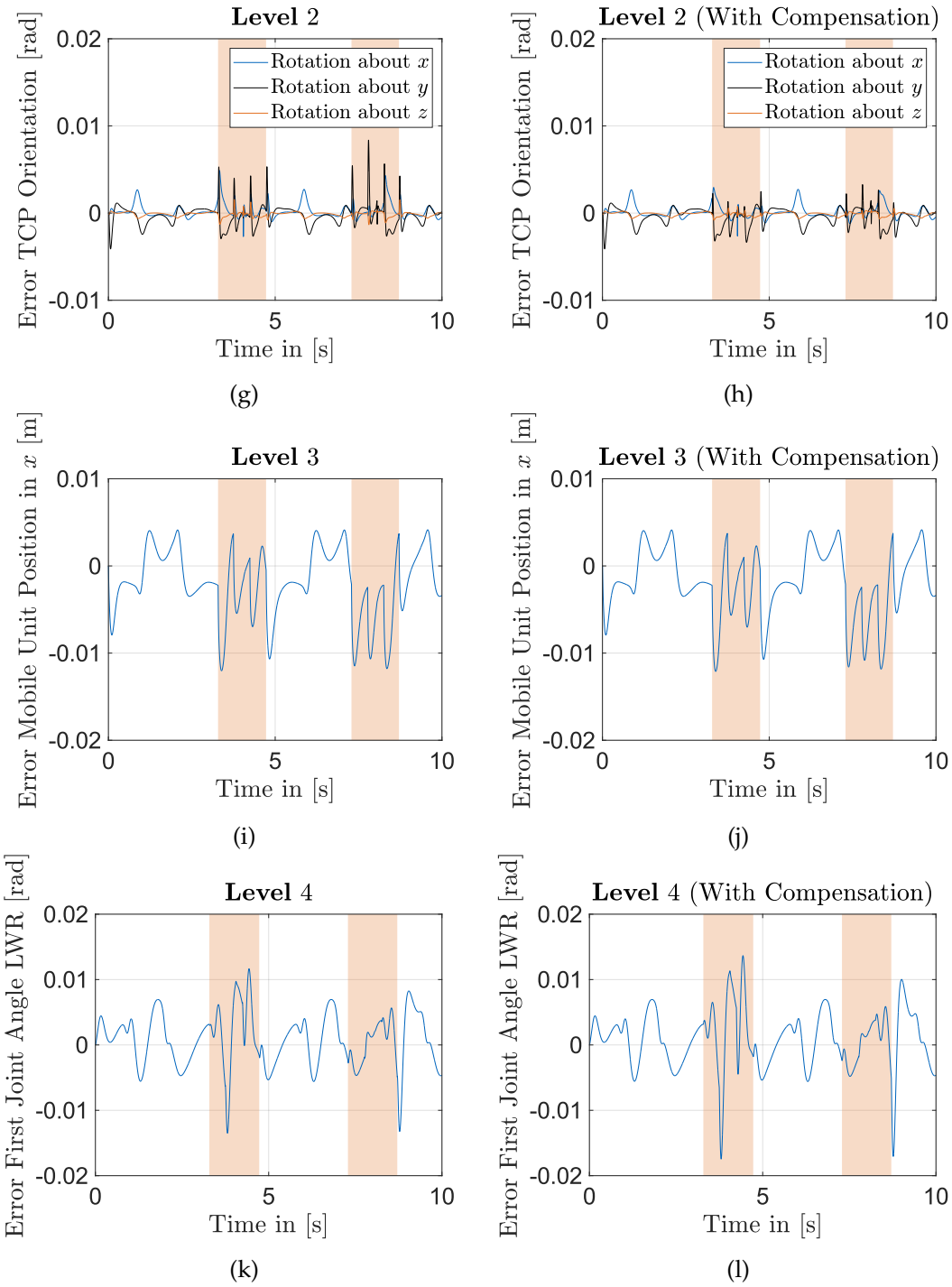
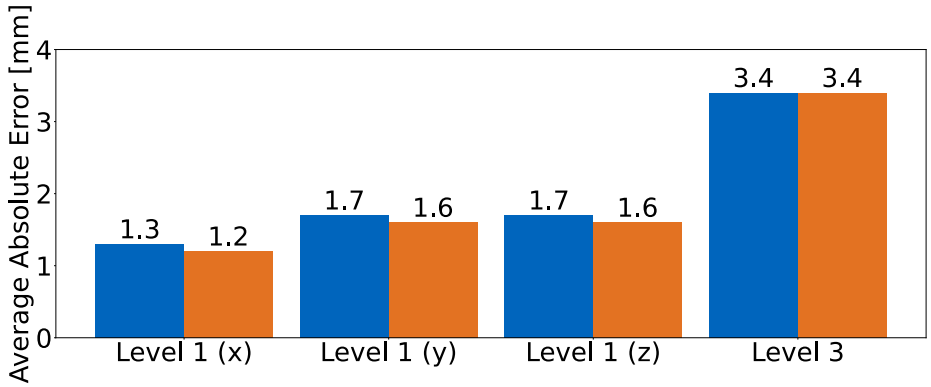
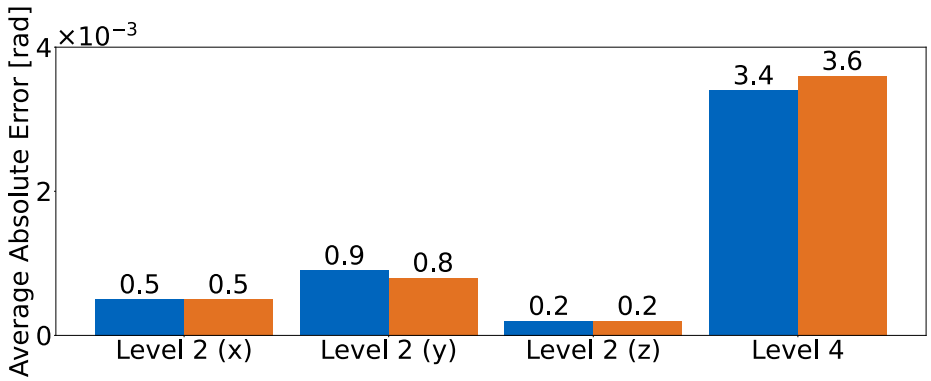


Figure 6.4.: Operational space errors for the hierarchical control without and with external force compensation.



(a) Average absolute errors for the tasks addressing translations of the robot.



(b) Average absolute errors for the tasks addressing rotations of the robot.

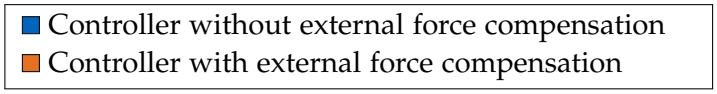
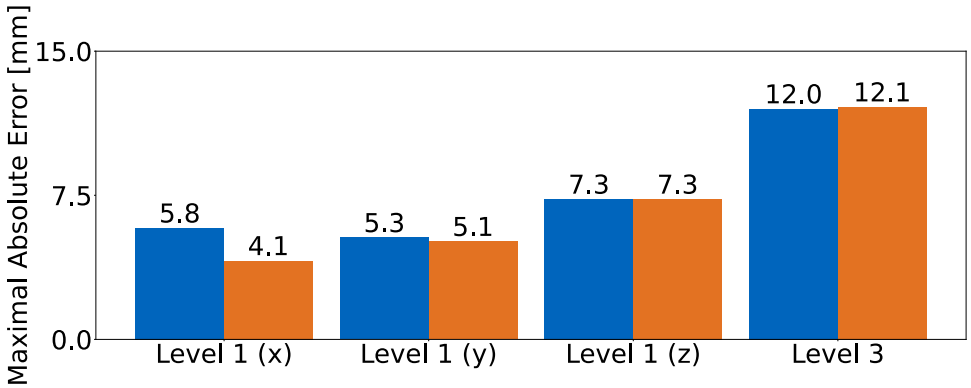
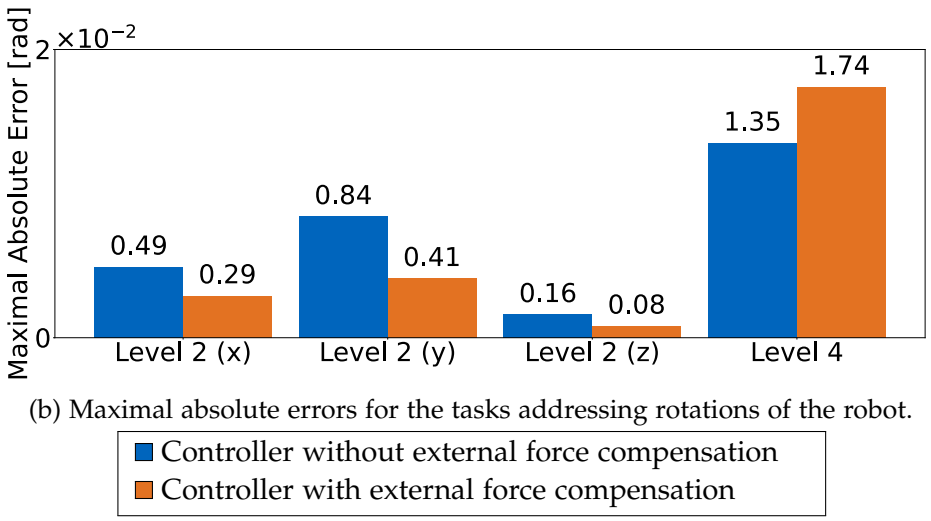


Figure 6.5.: The average absolute errors for every task pursued during the simulations.



(a) Maximal absolute errors for the tasks addressing translations of the robot.



(b) Maximal absolute errors for the tasks addressing rotations of the robot.

Figure 6.6.: The maximal absolute errors for every task pursued during the simulations.

7. Discussion

The implemented controller is capable of achieving good tracking behavior for all simulated scenarios. Moreover, the controller has shown to be able to react to disturbances, typically encountered in the setting of the SwarmRail system. These involve sudden position changes and high valued forces acting on the mobile unit. Both phenomena generally occur during the passing of a crossing point. Even without external force coupling compensation, our controller is able to adjust its joints quickly in order to reduce the error for the two highest prioritized tasks only. This is most notably visible for the TCPs deviation in x- and z-direction, for which the controller is able to reduce the error by 60% and 44%, respectively, compared to the corresponding deviation of the mobile unit. By additionally implementing the external force coupling compensation, we were able to reduce the deviation peaks caused by the collision forces even further for the two highest prioritized task levels. Achieving reductions up to a factor of 2 for the TCP position in x and rotation angle about z, compared to the controller without compensation. The sudden deviations caused by the mobile unit falling, however, were not further reduced by the compensation.

These observations were to be expected, given the nature of the external force. The external force acting on the mobile unit result from the impact in the x-direction. It is therefore applied on the task Level 3 of our defined hierarchy. All the other tasks do not present any external forces directly applied on them. That is

$$\boldsymbol{F}_{\dot{x}_i}^{\text{ext}} = \begin{cases} \mathbf{0} & \text{for } i \neq 3 \\ F_{\text{impact}} & \text{for } i = 3 \end{cases}, \tag{7.1}$$

for $i = 1 \dots 4$, with F_{impact} being defined according to (4.43).

The implemented compensation of external force couplings, eliminates the cross coupling between the external forces applied on a hierarchy Level i and all tasks of higher prioritization (refer to (5.37)). Applied to our case, the addition of a force compensation eliminates the coupling between the force applied on Level 3 and the tasks of Level 1 and 2. Consequently, the force compensation only adds terms influencing the upper two priority levels. Expressed differently, the lower priority levels do not benefit from adding a compensation term.

Furthermore, as only the collision forces were measured and compensated, the external force compensation was unable to reduce the error peak caused by the altitude drop. The controller only takes notice of the fall, by the altitude deviation caused by it. Therefore the tracking behavior for this disturbance is solely influenced by the same terms, which are also present for the uncompensated controller.

The discoveries provided by the simulated trajectories verify the controller's theoretical prospect of being applicable to the SwarmRail system. The main benefits of the SwarmRail concept compared to other overhead manipulators are given by the flexible and independent motion of the robots as well as the possibility of continuous overhead manipulation. In order to be able to make use of these advantages, the underlying whole-body controller must work reliably under all external conditions and disruptions which are expected during the movement inside the rail-network.

Nevertheless, it is important to classify this thesis' main objective appropriately. Our objective is to establish a possible guiding direction for the development of a whole-body controller for a physical demonstrator. Consequently, this thesis' theoretical benefit lies in giving an analysis regarding the controller's performance under extreme conditions, not yet implemented in the controller's original testing scope [6]. To which extend the results of this thesis are applicable to the real system remains to be investigated. For instance, let's take the effect achieved through external force compensation as an example: the effectiveness of said compensation term is owed to the fact, that we assumed exact measurement values of the external forces. Thus, this implies the existence of appropriate force torque sensors. However, high requirements regarding the accuracy of their respective measurements are being set. This imposes, hence, a major impediment for a possible real-world application [6].

Furthermore, the necessity of a force compensation term is also questionable for the physical demonstrator. As previously explained, the collision forces were deliberately chosen to be much higher than the forces expected for the demonstrator. Despite this, our uncompensated controller is already able to handle these large forces well. It also remains to be investigated, whether the force-torque cross coupling effects present at the demonstrator really justify the implementation of an external force coupling compensation. As it is pointed out in [6], the external force-torque cross couplings are usually small in practice, even during highly dynamic tasks.

On the other side, although subjected to high forces, the crossing point traversal of our model has been significantly simplified. Different to our model, the crossing points under real-world conditions allow the robot to additionally rotate and change the direction of translation. This introduces an additional degree of complexity to the problem, as the number of DOF increases from eight to ten. This is especially relevant should the centering by the PID controller be faulty. This would result in the robot entering the crossing point in a skewed manner. Thus, the mobile unit would use its full range of motion to reestablish the centered position inside the rail-network. Apart from the increased DOF, the collisions can also be expected to have a more severe impact on additional task levels.

Another factor to be considered when implementing this controller on the existing demonstrator is the choice of the admittance damping parameter. In this thesis, we chose the damping parameter to be equal to zero. This particular choice is made, in order to guarantee the applicability of the asymptotic stability proof by [6]. For the demonstrator, however, it is advised to include a damping parameter for the sake of

ensuring stability and robustness against phenomena not included in the model and uncertainties. Additionally, undesired oscillations of the robotic system can be avoided [70, 71].

8. Summary and Outlook

In this thesis, we adapted a hierarchical whole-body impedance tracking controller for the robots making up the SwarmRail system. Hence, our simulation model of the robots and the rail-network used for the evaluation of the controller, is built upon simplifying assumptions. Most notably we modeled the kinematic constraints set by the rails as a collection of high valued spring-dampers acting on the body. Using the singular perturbation approach, we justified the reduction from ten to eight DOF, respectively. By that, we ensured the omission of the mobile unit's two restricted motion directions for our controller.

Inspired by [7] our controller combines the kinematically-controlled mobile unit with a torque-controlled LWR, via the usage of an admittance interface. Based on this combined system, we implemented the hierarchical impedance-based tracking controller developed in [6]. By omitting the damping value for the admittance, we were thus able to adopt the derivations provided in [6]. By doing so, we concluded the asymptotic stability towards the equilibrium for said hierarchical tracking controller.

Having identified the passage of crossing points to be the most troublesome sections of the rail-network, we proceeded to simulate the robot's behavior while driving across bespoken sections. The traversing of these crossing points was modeled by four sudden and high valued collision forces, acting oppositely to the direction of translation. Additionally, the crossing point caused an altitude change of 5mm for the mobile unit. For the scope of this thesis, however, the passage of said characteristic points was observed with regards to a single translation direction for the mobile unit. Direction changes were not considered.

In a final step, the controller was validated inside a simulation. Throughout this simulation, a robot drove across the rails with the maximal velocity for which the reliable centering of the mobile unit could be guaranteed. What is more, for the sake of performance evaluation we distinguished between the controller with and without compensation of external force couplings. During our simulations we noticed, that both controllers were able to counteract the external disturbances. This was expressed by twofold means: the sudden error peaks were quickly reduced and position deviations at the TCP's position were only half as large as the corresponding deviations on the mobile unit.

However, we discovered that by including the external force coupling compensation an additional reduction of the maximal errors caused by the sudden disturbances was achieved. At the same time, the good overall tracking performance was maintained. Despite the theoretical performance benefits, we have acknowledged that the external force coupling compensation sets demanding conditions for our robotic system: at first,

we must guarantee the presence of the required force-torque sensors for the mobile unit. Secondly, all measurements provided by these sensors must fulfill high demands in their precision. This yields from the fact that the force compensation worked solely on the assumption of exactly knowing which forces acted on the robot. Hence, this is a requirement with strong limitations for real-world implementation [6].

The future work concerned with the SwarmRail system will be directed towards the practical implementation of the hierarchical whole-body impedance tracking controller presented in this thesis. However, before the actual control strategy can be considered, well known issues with the current demonstrator have to be addressed first. This includes guaranteeing reliable centering inside the rails under both, varying lighting conditions and rail surface properties. Additionally, the encountered issues regarding the wheels' loss of traction are yet to be solved. However, our group is currently working on the construction of a second demonstrator which intents to addresses all of the aforementioned shortcomings. Moreover, the second demonstrator has to enable the integration of a fully functional robotic arm at the bottom of the mobile unit.

Based on this new even more capable demonstrator, we are thus able to implement the control law depicted in this thesis. Just as it was the case for this thesis, the focus shall be set on the practical validation of the controller for the passage of crossing points. Performing tests on this demonstrator shall yield insights regarding the applicability of the controller for the system as a whole. Moreover, it shall provide information regarding additionally needed hardware components to be incorporated for the robot, such as force-torque sensors should the external force coupling compensation be deemed necessary. Should the tests conclude strong couplings between the different hierarchy levels, an adaptation of the controller as suggested by Wu et al. [19] could also be taken into consideration.

As a final note, a future challenge lies in defining a strategy for the crossing points, which takes all ten DOF into consideration. Throughout this thesis we assumed eight DOF to be available at all times, that is inside crossing points and straight segments. In reality, however, entering a crossing point allows the robot to make use of all its motion capabilities, having thus access to ten DOF. Concomitantly, performing a translation while inside the crossing point would return the mobile unit to a straight segment, consequently reducing the amount of DOF to eight once more. The fluent transition between control strategies for the straight segments and crossing points is not trivial and requires further investigation.

A. Mathematical Properties

A.1. Vector Orthogonality

Two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are orthogonal to each other if

$$\langle \boldsymbol{xy} \rangle = 0 \tag{A.1}$$

holds. Hereby, $\langle\rangle$ denotes the inner product [60].

A.2. Jacobian Matrix

Taken from [60]: for a vector $\boldsymbol{x}(\boldsymbol{q}) \in \mathbb{R}^m$, the corresponding Jacobian matrix is defined as

$$\boldsymbol{J}(\boldsymbol{x}) = \begin{bmatrix} \frac{\delta x_1}{\delta q_1}(\boldsymbol{q}) & \dots & \frac{\delta x_1}{\delta q_n}(\boldsymbol{q}) \\ \vdots & \ddots & \vdots \\ \frac{\delta x_m}{\delta q_1}(\boldsymbol{q}) & \dots & \frac{\delta x_m}{\delta q_n}(\boldsymbol{q}) \end{bmatrix} \in \mathbb{R}^{m \times n} \quad , \tag{A.2}$$

with x_i and q_i representing the ith element of vectors $\boldsymbol{x}$ and $\boldsymbol{q}$ respectively.

A.3. Skew Symmetry

Based on Petersen et. al [46]: a matrix A is said to be symmetric, if it is equal to its negative transpose:

$$\boldsymbol{A}^T = -\boldsymbol{A} \tag{A.3}$$

A.4. Positive Semidefiniteness

Based on Petersen et. al [46]: a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is said to be positive semidefinite, if for a all non-zero vector $\boldsymbol{x} \in \mathbb{R}^n$ the following property holds:

$$\boldsymbol{x}\boldsymbol{A}\boldsymbol{x}^T \geq 0 \quad .$$

A.5. Transpose of Multiplied Matrices

Taken from Karpfinger [60]: the transpose of the multiplication of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is given by:

$$(\boldsymbol{AB})^T = \boldsymbol{B}^T\boldsymbol{A}^T \tag{A.4}$$

A.6. Schur Complement

According to Boyd et al. [61], for a positive definite matrix

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^T & \boldsymbol{C} \end{bmatrix} \quad , \tag{A.5}$$

a positive definite blockmatrix $\boldsymbol{C}$ implies that:

$$\boldsymbol{A} - \boldsymbol{B}\boldsymbol{C}^{-1}\boldsymbol{B}^T \tag{A.6}$$

is also positive definite.

B. Mathematical Proofs

B.1. Proof of Equivalence

Following the derivation of section B.2, we can write

$$\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = -(\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}))^T \tag{B.1}$$

$$= -\dot{\boldsymbol{M}}(\boldsymbol{q})^T + 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T \Leftrightarrow \tag{B.2}$$

$$(\dot{\boldsymbol{M}} + \dot{\boldsymbol{M}}^T) = 2(\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T) \quad . \tag{B.3}$$

Due to the symmetry of the inertia matrix, the equation can be rewritten as

$$2\dot{\boldsymbol{M}} = 2(\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T) \Leftrightarrow \tag{B.4}$$

$$\dot{\boldsymbol{M}} = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T \quad . \tag{B.5}$$

B.2. Proof of Skew Symmetry

The proof for the skew symmetry (refer to (A.3)) of $\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, follows the steps laid out by Murray et al. [41].

$$(\dot{\boldsymbol{M}} - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}))_{i,j} = \dot{\boldsymbol{M}}_{i,j}(\boldsymbol{q}) - 2\boldsymbol{C}_{i,j}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{B.6}$$

$$= \sum_{k=1}^{n} \frac{\delta \boldsymbol{M}_{i,j}}{\delta \boldsymbol{q}_k}\dot{q}_k - \frac{\delta \boldsymbol{M}_{i,j}}{\delta \boldsymbol{q}_k}\dot{q}_k - \frac{\delta \boldsymbol{M}_{i,k}}{\delta \boldsymbol{q}_j}\dot{q}_k + \frac{\delta \boldsymbol{M}_{k,j}}{\delta \boldsymbol{q}_i}\dot{q}_k \tag{B.7}$$

$$= \sum_{k=1}^{n} \frac{\delta \boldsymbol{M}_{k,j}}{\delta q_i}\dot{q}_k - \frac{\delta \boldsymbol{M}_{i,k}}{\delta q_j}\dot{q}_k \tag{B.8}$$

By switching the indices i and j, and considering the symmetry of the inertia matrix, the following properties

$$(\dot{\boldsymbol{M}} - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}))_{j,i} = \sum_{k=1}^{n} \frac{\delta \boldsymbol{M}_{k,i}}{\delta \boldsymbol{q}_j}\dot{q}_k - \frac{\delta \boldsymbol{M}_{j,k}}{\delta \boldsymbol{q}_i}\dot{q}_k \tag{B.9}$$

$$= -\left(\sum_{k=1}^{n} \frac{\delta \boldsymbol{M}_{k,j}}{\delta \boldsymbol{q}_i}\dot{q}_k - \frac{\delta \boldsymbol{M}_{i,k}}{\delta \boldsymbol{q}_j}\dot{q}_k\right) \tag{B.10}$$

$$-(\dot{\boldsymbol{M}} - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}))_{i,j} \tag{B.11}$$

become evident.

B.3. Proof of Null Space Projector Consistency

Using the properties given by (3.9) for a operational space coordinate $\boldsymbol{x}_1 \in \mathbb{R}^m$ yields

$$\ddot{\boldsymbol{x}}_1 = \boldsymbol{J}_1(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_1(\boldsymbol{q})\dot{\boldsymbol{q}} \quad . \tag{B.12}$$

Rearranging this equation leads to

$$\boldsymbol{J}_1(\boldsymbol{q})\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{x}}_1 - \dot{\boldsymbol{J}}_1(\boldsymbol{q})\dot{\boldsymbol{q}} \quad . \tag{B.13}$$

This will be of importance in the derivations to come. Given the EOM

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}^{\text{ext}} + \left(\boldsymbol{\tau}_1 + \boldsymbol{\tau}_2^P\right) \quad , \tag{B.14}$$

multiplying these EOM with the inverse of the inertia matrix, namely $\boldsymbol{M}(\boldsymbol{q})^{-1}$, from the left side yields:

$$\ddot{\boldsymbol{q}} + \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{\tau}^{\text{ext}} + \boldsymbol{M}(\boldsymbol{q})^{-1}\left(\boldsymbol{\tau}_1 + \boldsymbol{\tau}_2^P\right) \quad . \tag{B.15}$$

Multiplication from the left side by $\boldsymbol{J}_1(\boldsymbol{q})$ and using the property expressed by (B.13) leads to

$$\ddot{\boldsymbol{x}}_1 - \dot{\boldsymbol{J}}_1(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{J}_1(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{J}_1(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{J}_1(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{\tau}^{\text{ext}} + \boldsymbol{J}_1(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\left(\boldsymbol{\tau}_1 + \boldsymbol{\tau}_2^P\right) \quad . \tag{B.16}$$

Introducing the function $\boldsymbol{p}$, which summarizes all terms which do not contain the acceleration expressed in the operational space or the projected torques of the second priority level, yields

$$\ddot{\boldsymbol{x}}_1 = \boldsymbol{p}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau}_1, \boldsymbol{\tau}^{\text{ext}}) + \boldsymbol{J}_1(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{\tau}_2^P \quad . \tag{B.17}$$

Using the definition of the projected torque provided by (3.23) leads to

$$\ddot{\boldsymbol{x}}_1 = \boldsymbol{p}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau}_1, \boldsymbol{\tau}^{\text{ext}}) + \boldsymbol{J}_1(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{N}_2^{\text{suc}}(\boldsymbol{q})\boldsymbol{\tau}_2 \quad . \tag{B.18}$$

Recapitulating the definition of dynamic consistency provided in section 3.4: joint torques linked to the null space projector do not cause accelerations for the higher priority task inside the operational space [21]. This equates to the requirement

$$\boldsymbol{J}_1(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{N}_2^{\text{suc}}(\boldsymbol{q})\boldsymbol{\tau}_2 \stackrel{!}{=} \boldsymbol{0} \quad . \tag{B.19}$$

For the following derivations, we omit the arguments of the dynamic parameters for the sake of clarity. That is: $\boldsymbol{M}(\boldsymbol{q}) = \boldsymbol{M}$, $\boldsymbol{J}_1(\boldsymbol{q}) = \boldsymbol{J}_1$ and $\boldsymbol{N}_2^{\text{suc}}(\boldsymbol{q}) = \boldsymbol{N}_2^{\text{suc}}$.

Using the definition of the null space projector (refer to (3.24)) and the generalized inverse (refer to (3.25)), results in the equations

$$\begin{aligned}\boldsymbol{J}_1\boldsymbol{M}^{-1}\boldsymbol{N}_2^{\text{suc}}\boldsymbol{\tau}_2 &= \boldsymbol{J}_1\boldsymbol{M}^{-1}\left(\boldsymbol{I} - \boldsymbol{J}_1^T(\boldsymbol{J}_1^{M+})^T\right)\boldsymbol{\tau}_2 \\ &= \boldsymbol{J}_1\boldsymbol{M}^{-1}\left(\boldsymbol{I} - \boldsymbol{J}_1^T\left(\boldsymbol{M}^{-1}\boldsymbol{J}_1^T(\boldsymbol{J}_1\boldsymbol{M}^{-1}\boldsymbol{J}_1^T)^{-1}\right)^T\right)\boldsymbol{\tau}_2 \quad .\end{aligned} \tag{B.20}$$

Using the symmetry of the inertia matrix and the rule expressed in section A.5 for the transpose of a matrix multiplication, the equation

$$\begin{aligned}\boldsymbol{J}_1\boldsymbol{M}^{-1}\boldsymbol{N}_2^{\text{suc}}\boldsymbol{\tau}_2 &= \boldsymbol{J}_1\boldsymbol{M}^{-1}\left(\boldsymbol{I} - \boldsymbol{J}_1^T(\boldsymbol{J}_1\boldsymbol{M}^{-1}\boldsymbol{J}_1^T)^{-1}\boldsymbol{J}_1\boldsymbol{M}^{-1}\right)\boldsymbol{\tau}_2 \\ &= \left(\boldsymbol{J}_1\boldsymbol{M}^{-1} - \underbrace{(\boldsymbol{J}_1\boldsymbol{M}^{-1}\boldsymbol{J}_1^T)(\boldsymbol{J}_1\boldsymbol{M}^{-1}\boldsymbol{J}_1^T)^{-1}}_{\boldsymbol{I}}\boldsymbol{J}_1\boldsymbol{M}^{-1}\right)\boldsymbol{\tau}_2 = (\boldsymbol{J}_1\boldsymbol{M}^{-1} - \boldsymbol{J}_1\boldsymbol{M}^{-1})\boldsymbol{\tau}_2 = \boldsymbol{0}\end{aligned} \tag{B.21}$$

can be derived.

Therefore, the torques of the lower prioritized task 2 do not affect the accelerations of the higher prioritizes task. The used null space projector is dynamically consistent.

B.4. Hierarchical Decoupling of Equations of Motion

Given are the EOM, derived in section 5.1. Those are

$$\boldsymbol{M}(\boldsymbol{y})\ddot{\boldsymbol{y}} + \boldsymbol{C}(\boldsymbol{y},\dot{\boldsymbol{y}})\dot{\boldsymbol{y}} + \boldsymbol{g}(\boldsymbol{y}) = \boldsymbol{\tau}^{\text{ext}} + \boldsymbol{\tau}^{\text{hier}} \quad . \tag{B.22}$$

In subsection 5.2.2, we already provided the relation relation between $\boldsymbol{v}$ and $\dot{\boldsymbol{q}}$. Since the stacked decoupled Jacobian is invertible, we can rewrite (5.19) as

$$\dot{\boldsymbol{y}} = \hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\boldsymbol{v} \quad . \tag{B.23}$$

Deriving this equation with respect to time, using the definition for the derivative of an inverse matrix given by Barned et al. [72], yields

$$\ddot{\boldsymbol{y}} = \hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\dot{\boldsymbol{v}} - \hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\dot{\hat{\boldsymbol{J}}}(\boldsymbol{y},\dot{\boldsymbol{y}})\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\boldsymbol{v} \quad . \tag{B.24}$$

Using these relations on the EOM leads to

$$\boldsymbol{M}(\boldsymbol{y})(\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\dot{\boldsymbol{v}} - \hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\dot{\hat{\boldsymbol{J}}}(\boldsymbol{y},\dot{\boldsymbol{y}})\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\boldsymbol{v}) + \boldsymbol{C}(\boldsymbol{y},\dot{\boldsymbol{y}})\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{y}) = \boldsymbol{\tau}^{\text{ext}} + \boldsymbol{\tau}^{\text{hier}} \quad \text{(B.25)}$$

$$\boldsymbol{M}(\boldsymbol{y})\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\dot{\boldsymbol{v}} + \left(\boldsymbol{C}(\boldsymbol{y},\dot{\boldsymbol{y}}) - \boldsymbol{M}(\boldsymbol{y})\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\dot{\hat{\boldsymbol{J}}}(\boldsymbol{y},\dot{\boldsymbol{y}})\right)\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{y}) = \boldsymbol{\tau}^{\text{ext}} + \boldsymbol{\tau}^{\text{hier}} \quad . \quad \text{(B.26)}$$

Multiplying by $\hat{\boldsymbol{J}}(\boldsymbol{y})^{-T}$ from the left side leads to

$$\underbrace{\hat{\boldsymbol{J}}(\boldsymbol{y})^{-T}\boldsymbol{M}(\boldsymbol{y})\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}}_{\boldsymbol{\Lambda}(\boldsymbol{y})}\dot{\boldsymbol{v}} + \underbrace{\hat{\boldsymbol{J}}(\boldsymbol{y})^{-T}\left(\boldsymbol{C}(\boldsymbol{y},\dot{\boldsymbol{y}}) - \boldsymbol{M}(\boldsymbol{y})\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}\dot{\hat{\boldsymbol{J}}}(\boldsymbol{y},\dot{\boldsymbol{y}})\right)\hat{\boldsymbol{J}}(\boldsymbol{y})^{-1}}_{\boldsymbol{\mu}(\boldsymbol{y},\dot{\boldsymbol{y}})}\boldsymbol{v} + \hat{\boldsymbol{J}}(\boldsymbol{y})^{-T}\boldsymbol{g}(\boldsymbol{y}) = \hat{\boldsymbol{J}}(\boldsymbol{y})^{-T}(\boldsymbol{\tau}^{\text{ext}} + \boldsymbol{\tau}^{\text{hier}}). \quad \text{(B.27)}$$

Rearranging this equation yields the EOM:

$$\boldsymbol{\Lambda}(\boldsymbol{y})\dot{\boldsymbol{v}} + \boldsymbol{\mu}(\boldsymbol{y},\dot{\boldsymbol{y}})\boldsymbol{v} = \hat{\boldsymbol{J}}(\boldsymbol{y})^{-T}\left(\boldsymbol{\tau}^{\text{ext}} + \boldsymbol{\tau}^{\text{hier}} - \boldsymbol{g}(\boldsymbol{y})\right) \quad , \quad \text{(B.28)}$$

which corresponds to (5.20) in subsection 5.2.2.

B.5. Equations of Motion in the Operational Space

For the following derivations, we omit the arguments of the dynamic parameters for the sake of clarity.

Using the relationship expressed in equations 5.24 and 5.25 for the hierarchically decoupled EOM (refer to (5.31)) yields

$$\boldsymbol{\Lambda}_i\left(\ddot{\boldsymbol{x}}_i + \sum_{j=1}^{i-1}\left(\boldsymbol{B}_{i,j}\ddot{\boldsymbol{x}}_j + \dot{\boldsymbol{B}}_{i,j}\dot{\boldsymbol{x}}_j\right)\right) + \boldsymbol{\mu}_{i,i}\left(\dot{\boldsymbol{x}}_i + \sum_{j=1}^{i-1}\boldsymbol{B}_{i,j}\dot{\boldsymbol{x}}_j\right) = \boldsymbol{F}_i^{\text{hier}} + \boldsymbol{F}_{\boldsymbol{v}_i}^{\text{ext}} \quad . \quad \text{(B.29)}$$

Rearranging this equation leads to:

$$\boldsymbol{\Lambda}_i\ddot{\boldsymbol{x}}_i + \left(\boldsymbol{\mu}_{i,i}\dot{\boldsymbol{x}}_i + \sum_{j=1}^{i-1}\left(\boldsymbol{\Lambda}_i\boldsymbol{B}_{i,j}\ddot{\boldsymbol{x}}_j + \boldsymbol{\Lambda}_i\dot{\boldsymbol{B}}_{i,j}\dot{\boldsymbol{x}}_j + \boldsymbol{\mu}_{i,i}\boldsymbol{B}_{i,j}\dot{\boldsymbol{x}}_j\right)\right) = \boldsymbol{F}_i^{\text{hier}} + \boldsymbol{F}_{\boldsymbol{v}_i}^{\text{ext}} \quad , \quad \text{(B.30)}$$

or expressed in matrix-vector notation:

$$\boldsymbol{\Lambda}_i\ddot{\boldsymbol{x}}_i + \boldsymbol{\mu}_{i,i}\dot{\boldsymbol{x}}_i + \begin{bmatrix}\boldsymbol{\mu}_{i,i}\boldsymbol{B}_{i,1} + \boldsymbol{\Lambda}_i\dot{\boldsymbol{B}}_{i,1} & \boldsymbol{\mu}_{i,i}\boldsymbol{B}_{i,2} + \boldsymbol{\Lambda}_i\dot{\boldsymbol{B}}_{i,2} & \dots & \boldsymbol{\mu}_{i,i}\boldsymbol{B}_{i,i-1} + \boldsymbol{\Lambda}_i\dot{\boldsymbol{B}}_{i,i-1}\end{bmatrix}\dot{\boldsymbol{x}}_{i-1}^{\text{aug}} + \begin{bmatrix}\boldsymbol{\Lambda}_i\boldsymbol{B}_{i,1} & \boldsymbol{\Lambda}_i\boldsymbol{B}_{i,2} & \dots & \boldsymbol{\Lambda}_i\boldsymbol{B}_{i,i-1}\end{bmatrix}\ddot{\boldsymbol{x}}_{i-1}^{\text{aug}} = \boldsymbol{F}_i^{\text{hier}} + \boldsymbol{F}_{\boldsymbol{v}_i}^{\text{ext}} \quad \text{(B.31)}$$

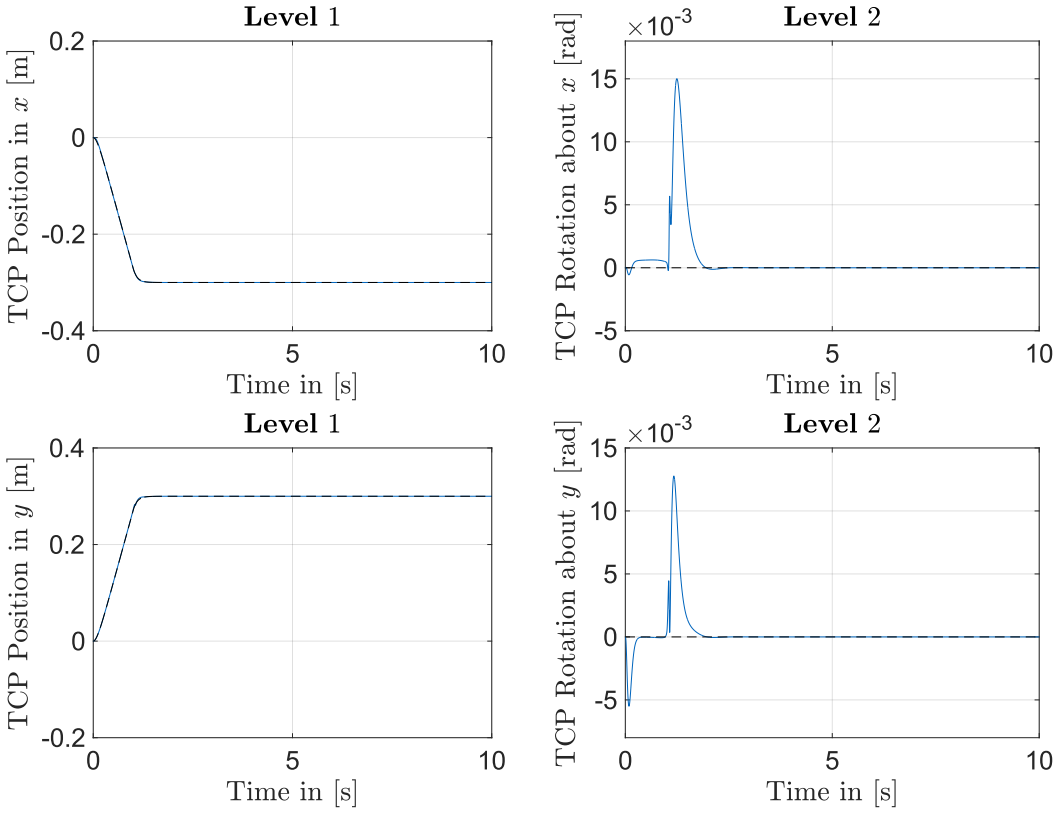
$$\boldsymbol{\Lambda}_i \ddot{\boldsymbol{x}}_i + \boldsymbol{\mu}_{i,i} \dot{\boldsymbol{x}}_i +$$
$$\underbrace{\begin{bmatrix} \boldsymbol{\mu}_{i,i}\boldsymbol{B}_{i,1} + \boldsymbol{\Lambda}_i \dot{\boldsymbol{B}}_{i,1} & \dots & \boldsymbol{\mu}_{i,i}\boldsymbol{B}_{i,i-1} + \boldsymbol{\Lambda}_i \dot{\boldsymbol{B}}_{i,i-1} & \boldsymbol{\Lambda}_i \boldsymbol{B}_{i,1} & \dots & \boldsymbol{\Lambda}_i \boldsymbol{B}_{i,i-1} \end{bmatrix}}_{\boldsymbol{\gamma}_i} \begin{bmatrix} \dot{\boldsymbol{x}}_{i-1}^{\text{aug}} \\ \ddot{\boldsymbol{x}}_{i-1}^{\text{aug}} \end{bmatrix} = \boldsymbol{F}_i^{\text{hier}} + \boldsymbol{F}_{v_i}^{\text{ext}}.$$

C. Simulations

This chapter provides additional simulation results which did not provide relevant insights regarding the robustness of our controller when exposed to strong disturbances. Nevertheless, these simulations are kept in the Appendix for the interested reader.

C.1. Linear Trajectories for the Rotational Scenario

Our controller is able to follow the desired trajectory well. After an initial overshoot, the error converges to zero and the robot maintains it's configuration.



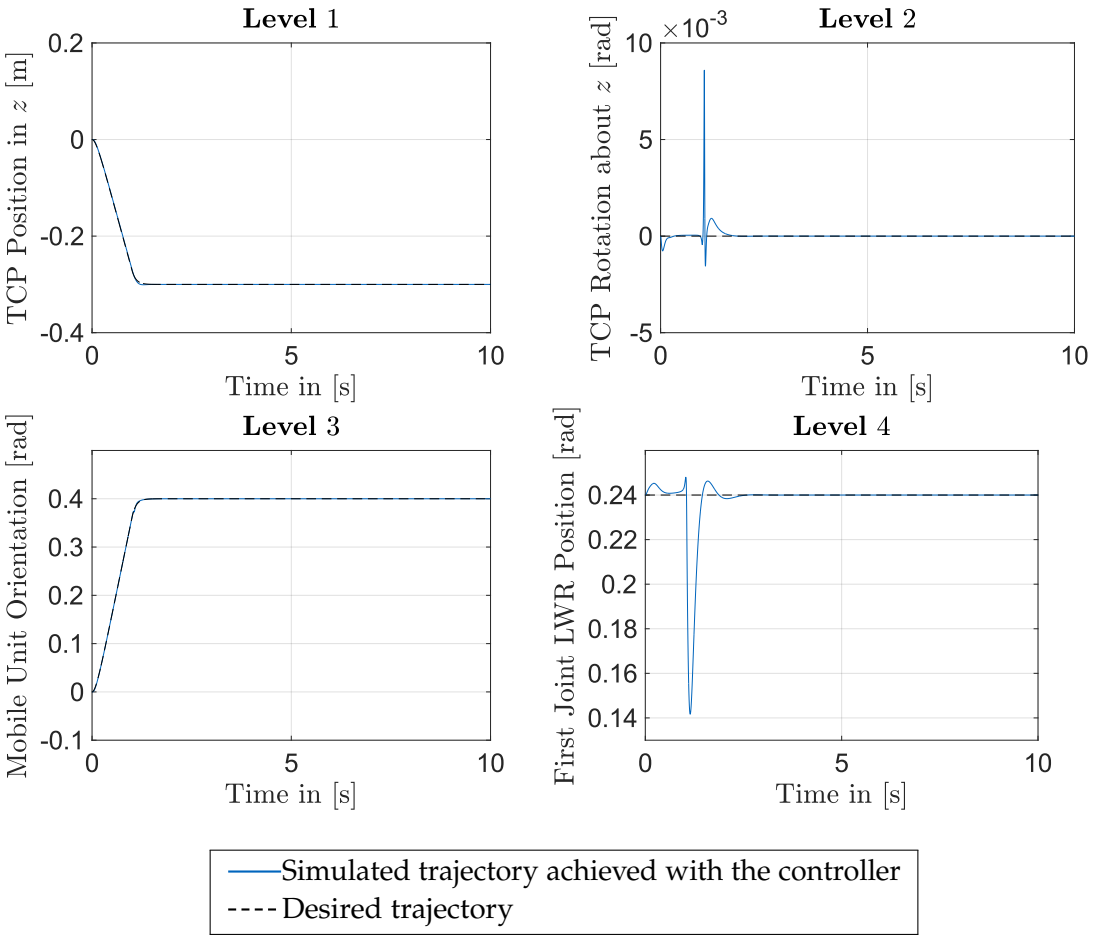
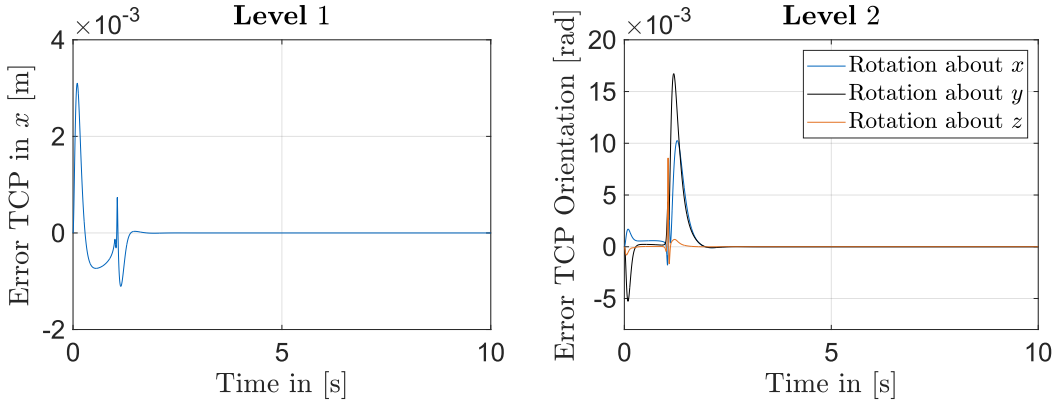


Figure C.1.: Trajectories for each level for the RS tracking linear trajectories.



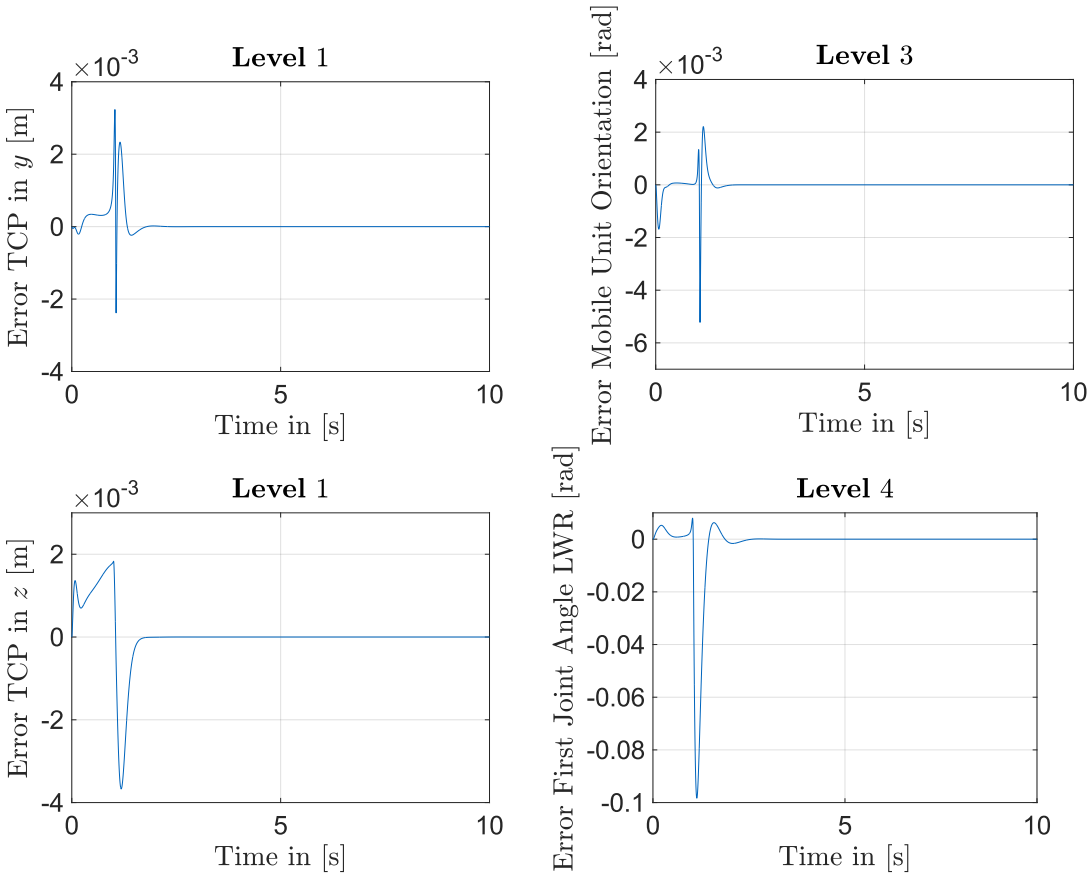
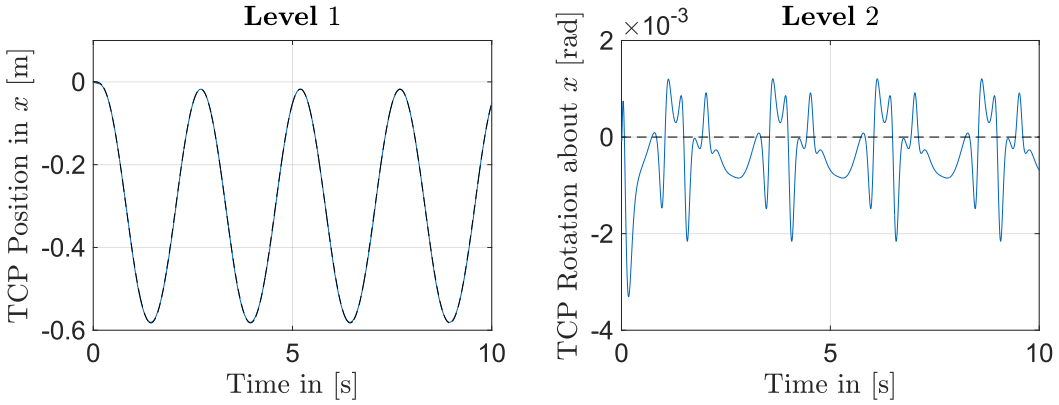


Figure C.2.: Tracking errors for each level for the RS tracking linear trajectories.

C.2. Sinusoidal Trajectories for the Rotational Scenario

Our controller is able to follow the desired trajectory well, with the errors commuting into small ranges.



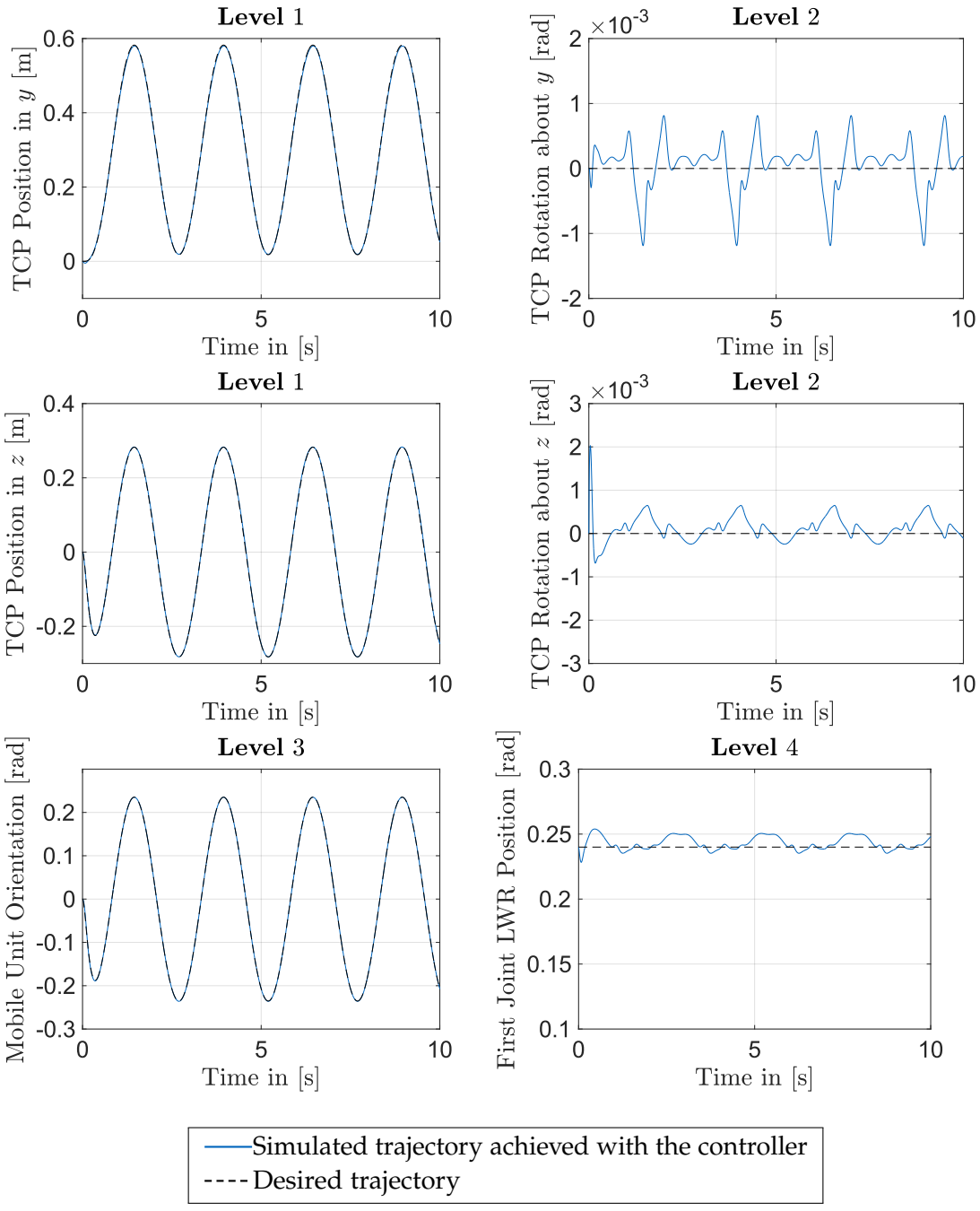


Figure C.3.: Trajectories for each level for the RS tracking sinusoidal trajectories.

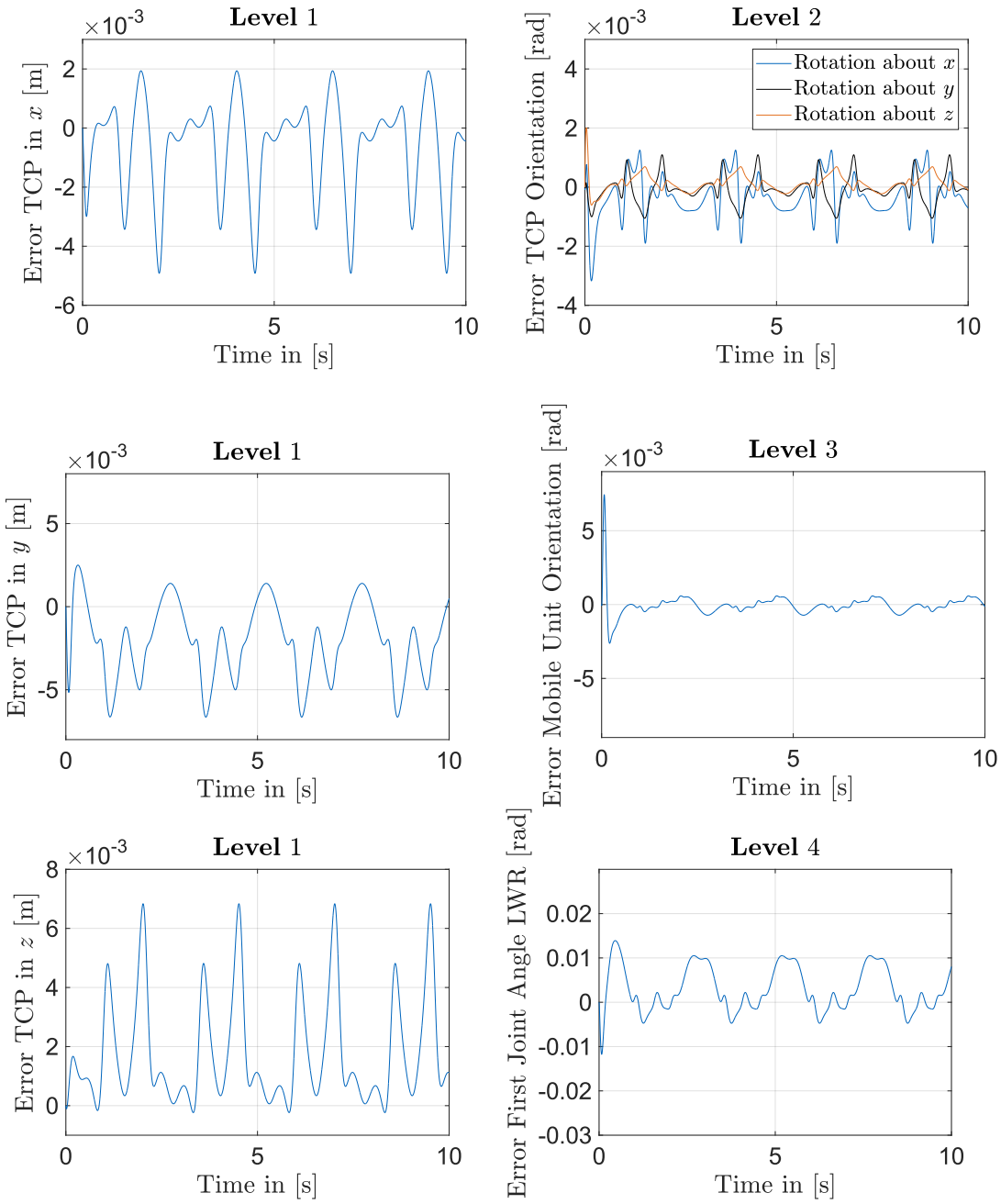
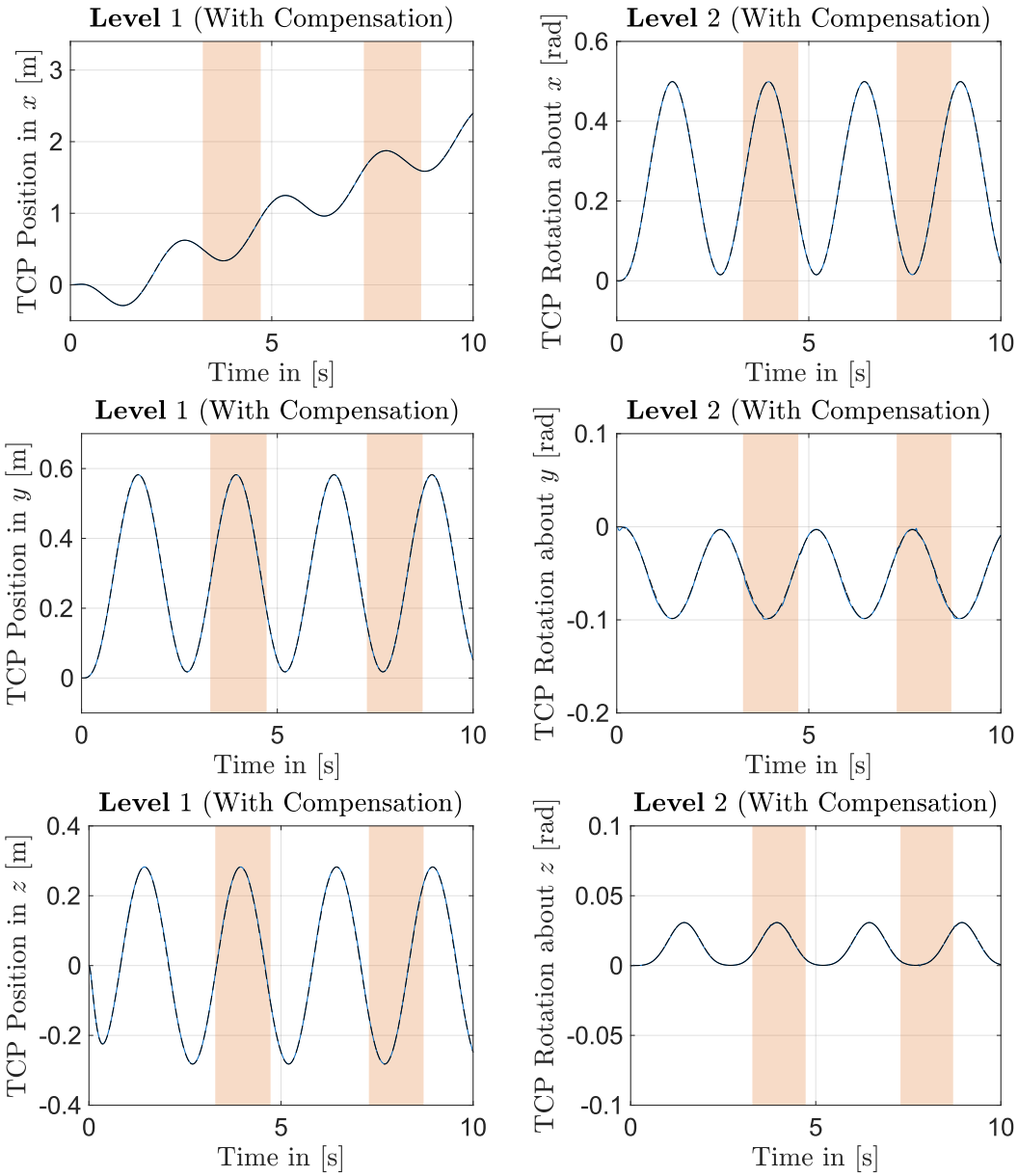


Figure C.4.: Tracking errors for each level for the RS tracking sinusoidal trajectories.

C.3. Sinusoidal Trajectories for the Translational Scenario with Compensation

Our controller with implemented external force coupling compensation is able to follow the desired trajectory well. The effects of the crossing point traversion are barely visible.


Level 1 (With Compensation)
TCP Position in x [m]
Time in [s]
Level 2 (With Compensation)
TCP Rotation about x [rad]
Time in [s]
Level 1 (With Compensation)
TCP Position in y [m]
Time in [s]
Level 2 (With Compensation)
TCP Rotation about y [rad]
Time in [s]
Level 1 (With Compensation)
TCP Position in z [m]
Time in [s]
Level 2 (With Compensation)
TCP Rotation about z [rad]
Time in [s]

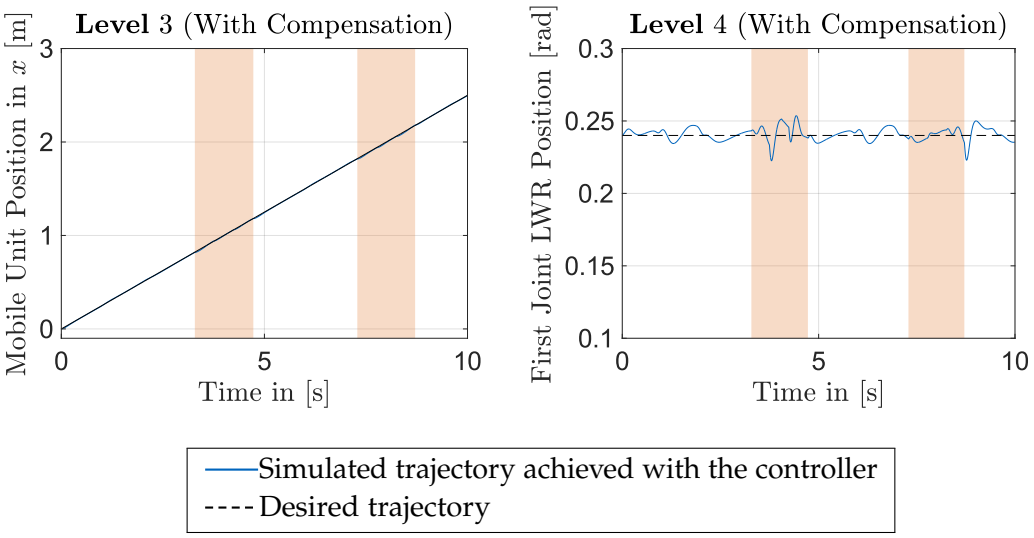


Figure C.5.: Tracked trajectories for the controller using external coupling force compensation.

List of Figures

List of Tables

Acronyms

COM Center of Mass. 27, 29, 38–40

DC Direct-Current. 6

DLR German Aerospace Center. 1, 4, 6, 7, 39, 41, 85

DOF Degree(s) of Freedom. 3, 16, 18, 22, 25, 27–32, 35–38, 41, 52, 55, 64, 67, 68, 85, 86

EOM Equations of Motion. 3, 18, 19, 33, 35–37, 41–43, 46–50, 72–74

IMU Inertial Measurement Unit. 9, 54

IR Infrared. 7, 8

LWR Lightweight Robot. 7, 26, 28, 33, 37, 41–43, 52, 57, 67

MFZ Multifunctional Cell. 4, 10

OSF Operational Space Formulation. 3

PID Proportional–Integral–Derivative. 8, 10, 11, 64

RS Rotational Scenario. 37, 38, 51, 52, 78–81, 86

TCP Tool Center Point. 17, 20, 21, 52, 55, 57, 58, 63, 67, 85

TS Translational Scenario. 37, 38, 51–53

Bibliography

[1] I. Karabegović, E. Karabegović, M. Mahmić, and E. Husak. "The application of service robots for logistics in manufacturing processes." In: *Advances in Production Engineering & Management* 10.4 (2015).

[2] G. Radivojević and L. Milosavljević. "The concept of logistics 4.0". In: *4th Logistics International Conference*. 2019, pp. 23–25.

[3] *Factory of the future: Intelligent robots for digitally-driven production*. Accessed: 2023-04-27. URL: %5Curl%7Bhttps://factory-of-the-future.dlr.de/%7D.

[4] M. Görner, F. Benedikt, F. Grimmel, and T. Hulin. "SwarmRail: A Novel Overhead Robot System for Indoor Transport and Mobile Manipulation". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 5905–5911.

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

[6] A. Dietrich and C. Ott. "Hierarchical impedance-based tracking control of kinematically redundant robots". In: *IEEE Transactions on Robotics* 36.1 (2019), pp. 204–221.

[7] A. Dietrich, K. Bussmann, F. Petit, P. Kotyczka, C. Ott, B. Lohmann, and A. Albu-Schäffer. "Whole-body impedance control of wheeled mobile manipulators: Stability analysis and experiments on the humanoid robot Rollin'Justin". In: *Autonomous Robots* 40 (2016), pp. 505–517.

[8] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2010. ISBN: 1849966346.

[9] A. Dietrich, C. Ott, and A. Albu-Schäffer. "An overview of null space projections for redundant, torque-controlled robots". In: *The International Journal of Robotics Research* 34.11 (2015), pp. 1385–1400.

[10] S. Fahmi, C. Mastalli, M. Focchi, and C. Semini. "Passive whole-body control for quadruped robots: Experimental validation over challenging terrain". In: *IEEE Robotics and Automation Letters* 4.3 (2019), pp. 2553–2560.

[11] K. Bouyarmane and A. Kheddar. "On weight-prioritized multitask control of humanoid robots". In: *IEEE Transactions on Automatic Control* 63.6 (2017), pp. 1632–1647.

[12] N. Dehio, R. F. Reinhart, and J. J. Steil. "Multiple task optimization with a mixture of controllers for motion generation". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 6416–6421.

[13] J. Salini, V. Padois, and P. Bidaud. "Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions". In: *2011 IEEE international conference on robotics and automation*. IEEE. 2011, pp. 1283–1290.

[14] F. L. Moro, M. Gienger, A. Goswami, N. G. Tsagarakis, and D. G. Caldwell. "An attractor-based whole-body motion control (WBMC) system for humanoid robots". In: *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE. 2013, pp. 42–49.

[15] A. Escande, N. Mansard, and P.-B. Wieber. "Hierarchical quadratic programming: Fast online humanoid-robot motion generation". In: *The International Journal of Robotics Research* 33.7 (2014), pp. 1006–1028.

[16] J. Baillieul, J. Hollerbach, and R. Brockett. "Programming and control of kinematically redundant manipulators". In: *The 23rd IEEE Conference on Decision and Control*. IEEE. 1984, pp. 768–774.

[17] Y. Nakamura, H. Hanafusa, and T. Yoshikawa. "Task-priority based redundancy control of robot manipulators". In: *The International Journal of Robotics Research* 6.2 (1987), pp. 3–15.

[18] J. Hollerbach and K. Suh. "Redundancy resolution of manipulators through torque optimization". In: *IEEE Journal on Robotics and Automation* 3.4 (1987), pp. 308–316.

[19] X. Wu, C. Ott, A. Albu-Schäffer, and A. Dietrich. "Passive Decoupled Multitask Controller for Redundant Robots". In: *IEEE Transactions on Control Systems Technology* 31.1 (2022), pp. 1–16.

[20] O. Khatib. "A unified approach for motion and force control of robot manipulators: The operational space formulation". In: *IEEE Journal on Robotics and Automation* 3.1 (1987), pp. 43–53.

[21] O. Khatib. "Inertial properties in robotic manipulation: An object-level framework". In: *The international journal of robotics research* 14.1 (1995), pp. 19–36.

[22] L. Sentis and O. Khatib. "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives". In: *International Journal of Humanoid Robotics* 2.04 (2005), pp. 505–518.

[23] L. Sentis, J. Park, and O. Khatib. "Compliant control of multicontact and center-of-mass behaviors in humanoid robots". In: *IEEE Transactions on robotics* 26.3 (2010), pp. 483–501.

[24] O. Khatib, L. Sentis, J. Park, and J. Warren. "Whole-body dynamic behavior and control of human-like robots". In: *International Journal of Humanoid Robotics* 1.01 (2004), pp. 29–43.

[25] C. Ott, A. Kugi, and Y. Nakamura. "Resolving the problem of non-integrability of nullspace velocities for compliance control of redundant manipulators by using semi-definite lyapunov functions". In: *2008 IEEE international conference on robotics and automation*. IEEE. 2008, pp. 1999–2004.

[26] N. Hogan. "Impedance Control: An Approach to Manipulation: Part I—Theory". In: *Journal of Dynamic Systems, Measurement, and Control* 107.1 (1985), p. 1. DOI: 10.1115/1.3140702. URL: https://doi.org/10.1115%5C%2F1.3140702.

[27] N. Hogan. "Impedance Control: An Approach to Manipulation: Part II—Implementation". In: *Journal of Dynamic Systems, Measurement, and Control* 107.1 (1985), p. 8. DOI: 10.1115/1.3140713. URL: https://doi.org/10.1115%2F1.3140713.

[28] N. Hogan. "Impedance Control: An Approach to Manipulation: Part III—Applications". In: *Journal of Dynamic Systems, Measurement, and Control* 107.1 (1985), p. 17. DOI: 10.1115/1.3140701. URL: https://doi.org/10.1115%2F1.3140701.

[29] E. Ackerman. "Toyota Research Demonstrates Ceiling-Mounted Home Robot". In: *IEEE Spectrum, Sep* (2020).

[30] F. Krebs, L. Larsen, G. Braun, and W. Dudenhausen. "Design of a multifunctional cell for aerospace CFRP production". In: *The International Journal of Advanced Manufacturing Technology* 85 (2016), pp. 17–24.

[31] M. Wulfraat. *Swisslog Autostore: an in-depth review of automated split case picking technology for distribution centers*. 2019.

[32] M. Görner, O. Eiberger, T. Hulin, K. Nottensteiner, and R. Weitschat. "Forschung für die Fabrik der Zukunft-Flexibilität durch Robotik". In: *Technik in Bayern, Ausgabe Süd* 2021.02 (2021), pp. 22–23.

[33] J. Dietrich, G. Hirzinger, B. Gombert, and J. Schott. "On a unified concept for a new generation of light-weight robots". In: *Experimental Robotics I: The First International Symposium Montreal, June 19–21, 1989*. Springer. 2006, pp. 287–303.

[34] G. Hirzinger, A. Baader, R. Koeppe, and M. Schedl. "Towards a new generation of multisensory light-weight robots with learning capabilities". In: *IFAC Proceedings Volumes* 26.2 (1993), pp. 665–672.

[35] B. Gombert, G. Hirzinger, G. Plank, M. Schedl, and J. Shi. "Modular concepts for the new generation of DLR's light weight robots". In: *Proceedings of the Third Conference on Mechatronics and Robotics: "From design methods to industrial applications"*. Springer. 1995, pp. 30–43.

[36] *SwarmRail*. Accessed: 2023-04-12. URL: %5Curl%7Bhttps://www.dlr.de/rm/en/desktopdefault.aspx/tabid-13938/#gallery/36115%7D.

[37] S. Zacher and M. Reuter. *Regelungstechnik für Ingenieure*. Springer, 2017.

[38] F. Grimmel. "Programmierung eines Mikrocontrollers zur Ansteuerung mobiler robotischer Einheiten für das SwarmRail-System". PhD thesis. Hochschule für angewandte Wissenschaften München, 2019.

[39] A. Dietrich, X. Wu, K. Bussmann, M. Harder, M. Iskandar, J. Englsberger, C. Ott, and A. Albu-Schäffer. "Practical consequences of inertia shaping for interaction and tracking in robot control". In: *Control Engineering Practice* 114 (2021), p. 104875.

[40] 1. Craig. *Introduction to robotics : mechanics & control / John J. Craig.* Includes bibliographies and index. Reading, Mass.: Addison-Wesley Pub. Co., 1986. ISBN: 0201103265.

[41] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[42] *Cartesian coordinates*. DOI: 10.1093/oi/authority.20110803095552345. URL: https://www.oxfordreference.com/view/10.1093/oi/authority.20110803095552345.

[43] M. Day and I. Games. "Extracting euler angles from a rotation matrix". In: *Insomniac Games R&D. Available online at: http://www. insomniacgames. com/mike-day-extracting-eulerangles-from-a-rotation-matrix* (2012).

[44] C. Ott. *Cartesian impedance control of redundant and flexible-joint robots*. Springer, 2008.

[45] M. H. Ang, O. Khatib, and B. Siciliano. *Encyclopedia of Robotics*. Springer Berlin Heidelberg, 2019.

[46] K. B. Petersen, M. S. Pedersen, et al. "The matrix cookbook". In: *Technical University of Denmark* 7.15 (2008), p. 510.

[47] X. Wu. "Passivation of a hierarchical whole-body controller for humanoid robots". PhD thesis. Technische Universität München, 2016.

[48] S. Haddadin. *Towards safe robots: approaching Asimov's 1st law*. Vol. 90. Springer, 2013.

[49] S. B. Slotine and B. Siciliano. "A general framework for managing multiple tasks in highly redundant robotic systems". In: *proceeding of 5th International Conference on Advanced Robotics*. Vol. 2. 1991, pp. 1211–1216.

[50] P. Baerlocher and R. Boulic. "Task-priority formulations for the kinematic control of highly redundant articulated structures". In: *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190)*. Vol. 1. IEEE. 1998, pp. 323–329.

[51] T. B. Wimböck. "Controllers for compliant two-handed dexterous manipulation". PhD thesis. Technische Universität Wien, 2012.

[52] C. Ott, A. Albu-Schaffer, A. Kugi, S. Stamigioli, and G. Hirzinger. "A passivity based cartesian impedance controller for flexible joint robots-part I: Torque feedback and gravity compensation". In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004.* Vol. 3. IEEE. 2004, pp. 2659–2665.

[53] A. Albu-Schaffer, C. Ott, and G. Hirzinger. "A passivity based cartesian impedance controller for flexible joint robots-part II: Full state feedback, impedance design and experiments". In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*. Vol. 3. IEEE. 2004, pp. 2666–2672.

[54] T. Wimböck and C. Ott. "Dual-arm manipulation". In: *Towards Service Robots for Everyday Environments: Recent Advances in Designing Service Robots for Complex Tasks in Everyday Environments*. Springer, 2012, pp. 353–366.

[55] P. Kokotovic and P. Sannuti. "Singular perturbation method for reducing the model order in optimal control design". In: *IEEE Transactions on Automatic Control* 13.4 (1968), pp. 377–384.

[56] P. Sannuti. *Singular perturbation method in the theory of optimal control*. University of Illinois at Urbana-Champaign, 1968.

[57] P. Sannuti and P. Kokotovic. "Near-optimum design of linear systems by a singular perturbation method". In: *IEEE Transactions on Automatic Control* 14.1 (1969), pp. 15–22.

[58] C. Lobry and T. Sari. "Singular perturbation methods in control theory". In: *Contrôle non linéaire et Applications* 64 (2005), pp. 155–182.

[59] K. Magnus, K. Popp, W. Sextro, K. Magnus, K. Popp, and W. Sextro. "Grundbegriffe und Darstellungsmittel". In: *Schwingungen: Grundlagen–Modelle–Beispiele* (2021), pp. 1–20.

[60] C. Karpfinger. *Höhere Mathematik in Rezepten*. Springer, 2022.

[61] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press Cambridge, 2004.

[62] L. L. Whitcomb, A. A. Rizzi, and D. E. Koditschek. "Comparative experiments with a new adaptive controller for robot arms". In: *IEEE Transactions on Robotics and Automation* 9.1 (1993), pp. 59–70.

[63] C. Borst, T. Wimbock, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. R. Giordano, R. Konietschke, W. Sepp, S. Fuchs, et al. "Rollin'justin-mobile platform with variable base". In: *2009 IEEE International Conference on Robotics and Automation*. IEEE. 2009, pp. 1597–1598.

[64] K. Bussmann. "Whole-Body Stability for a Humanoid Robot: Analysis, Control Design, and Experimental Evaluation". PhD thesis. Technische Universität München, 2014.

[65] A. D. Luca and G. Oriolo. *Modelling and control of nonholonomic mechanical systems*. Springer, 1995.

[66] M. Iskandar, O. Eiberger, A. Albu-Schäffer, A. De Luca, and A. Dietrich. "Collision detection, identification, and localization on the dlr sara robot with sensing redundancy". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 3111–3117.

[67] T. M. Inc. *MATLAB version: 9.13.0 (R2022b)*. Natick, Massachusetts, United States, 2022. URL: https://www.mathworks.com.

[68] S. Documentation. *Simulation and Model-Based Design*. 2020. URL: https://www.mathworks.com/products/simulink.html.

[69] A. Albu-Schaffer, C. Ott, U. Frese, and G. Hirzinger. "Cartesian impedance control of redundant robots: Recent results with the DLR-light-weight-arms". In: *2003 IEEE International conference on robotics and automation (Cat. No. 03CH37422)*. Vol. 3. IEEE. 2003, pp. 3704–3709.

[70] D. Bazzi, M. Lapertosa, A. M. Zanchettin, and P. Rocco. "Goal-driven variable admittance control for robot manual guidance". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 9759–9766.

[71] A. Gmerek and E. Jezierski. "Admittance control of a 1-DoF robotic arm actuated by BLDC motor". In: *2012 17th International Conference on Methods & Models in Automation & Robotics (MMAR)*. IEEE. 2012, pp. 633–638.

[72] R. J. Barnes. "Matrix differentiation". In: *Springs Journal* (2006), pp. 1–9.