Mobile Robot Manipulator Control for Interacting with 1-DoF Mechanisms

Yuyao Liu

LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6299
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2025

Abstract

This study investigates the interaction problem between a differential-drive mobile manipulator and one-degree-of-freedom (1-DoF) mechanisms. While previous research has focused on stationary robotic platforms, this work puts emphasis on mobile interaction.

Building upon a previously proposed adaptive control strategy for fixed-base manipulators, a holistic control framework is developed to coordinate the motion of the mobile base and the manipulator. The adaptive control strategy is extended to mobile robots, enabling repositioning during interaction to achieve larger door opening angles and avoid collisions. A nonlinear velocity controller is designed to guide the differential-drive mobile base to a suitable position for initiating interaction. To achieve simultaneous motion between the mobile base and the manipulator, an extended Jacobian matrix formulation is employed, integrating both forward and inverse kinematics of the base and arm.

To maintain a safe distance between the manipulator and the mobile base, a null-space optimization method is integrated. By modifying the base orientation in accordance with the direction of motion, this method also compensates for the non-holonomic restrictions of differential-drive platforms. Additionally, to improve manipulability and prevent kinematic singularities during the door-opening job, an adaptive damping factor is incorporated. Furthermore, building upon previously recorded motion trajectories, an online estimator is introduced, allowing the robot to learn the rotational axis, the door's rotation radius, and the expected motion direction.

Simulation and real robot experiments are conducted to evaluate the suggested control framework in a range of door-opening scenarios. The results demonstrate that the proposed framework achieves robust coordination between the mobile base and manipulator, enabling safe and efficient

interaction with various 1-DoF mechanisms. The findings confirm the applicability of the approach for real-world mobile manipulation tasks under kinematic uncertainties.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Yiannis for providing me with such an interesting topic at the RobotLab—after all, who can say no to a robot, the ultimate grown-up toy? From choosing the topic and outlining the structure to exploring related work, he offered tremendous support throughout the process. My project is also based on research he was previously involved in. During our weekly discussions, he consistently guided me toward the next steps and patiently answered my questions. I also deeply appreciate his tolerance of my chronic procrastination.

I would also like to thank Marko. Since the autumn semester of the "FRTN70 - Project in Systems, Control and Learning" course, I have been using the control libraries he developed to control the UR5e robot arm. His YouTube tutorials taught me how to use Docker, Pinocchio, and other essential tools. Whether it was setting up the environment or writing control programs, he provided invaluable assistance. He was always willing to spend hours answering my naive questions and gave incredibly detailed responses. The world cannot live without him.

My thanks also go to Björn. It was he whom I first contacted via email to ask whether there were any Master thesis projects available in the RobotLab. Passing by his office and seeing his cheerful smile always brightened my day.

I am grateful to everyone who contributed to the maintenance of the Heron robot control code. I would also like to thank my classmates and friends who made my time as a graduate student less lonely and more enjoyable. Finally, I want to thank the RobotLab for providing a dedicated project room. I truly love this space—especially how convenient the restrooms are.

Contents

1

Introduction

1.1 Background

In both domestic and industrial environments, robots are frequently required to interact with mechanisms that have one degree of freedom (1-DoF), such as doors, drawers, and windows. These mechanisms are ubiquitous in homes, offices, hospitals, and factories, serving essential functions in accessing storage, partitioning spaces, and securing environments. Effective and autonomous interaction with such mechanisms is a prerequisite for service robots to operate robustly in unstructured human environments.

However, a significant challenge arises from the fact that the kinematic parameters of these mechanisms—such as the orientation of the rotational axis, the rotation radius, or whether the joint is prismatic or revolute—are typically unknown beforehand. Conventional sensing approaches, such as visual perception or manual modeling, often fall short because of occlusions, sensor noise, or environmental variability. These uncertainties make it difficult for robots to plan appropriate motion trajectories and ensure safe and efficient physical interaction.

While many researchers have explored the challenges of interacting with unknown single-DOF mechanisms, most assume that the manipulator is fixed [1] [2]. This simplification ignores the mobility of service robots, which is critical when the target mechanism is distributed across different locations. The inclusion of mobility increases the complexity of planning and control as the robot must simultaneously manage whole-body coordination, base positioning, and safe interaction with the mechanism. In addition, the motion of the mobile base must obey non-holonomic constraints and limited actuation capabilities, which pose further challenges for reliable and adaptive interaction. This thesis focuses on extending the interaction strategy to mobile manipulators to address the unique difficul-

ties and opportunities that arise when the robot needs to approach, align, and manipulate single-DOF mechanisms during motion.

In this thesis, we aim to extend the adaptive control framework originally proposed for fixed-base manipulators by Karayiannidis et al. [3] to mobile manipulators, thereby enabling a differential-drive platform to *(i)* reposition itself during the task, *(ii)* exploit arm-base coordination to enlarge the attainable workspace, and *(iii)* estimate the mechanism's kinematic properties online during the interaction. Compared with fixed bases, mobile platforms introduce non-holonomic constraints and underactuation that fundamentally change the inverse kinematics problem and call for a holistic whole-body approach.

The structure of this thesis is organized as follows: In the next section, we provide an overview of related work on mobile manipulators. Chapter 2 presents the kinematic model of the system and formulates the problem. Chapter 3 introduces the strategy for guiding the mobile base to a designated position in preparation for door manipulation. The proposed whole-body control method and an optimization approach for ensuring safe and smooth motion are detailed in Chapter 4, followed by an online estimator developed to learn the kinematic constraint parameters during interaction in Chapter 5. Chapter 6 evaluates the performance of the overall control framework, and concluding remarks are provided in Chapter 7.

1.2 Related Work and Objectives

Research on robotic door-opening has converged on two complementary themes: *(i)* perceiving and modeling the unknown kinematic constraints, and *(ii)* executing compliant motions that respect those constraints while remaining safe and efficient.

Niemeyer and Slotine proposed a minimization strategy for interacting with a door without prior knowledge of its kinematic structure [4]. Their approach exploits the "follow the path of least resistance" principle, allowing the mechanism to correct its direction of motion during interaction. The control framework consists of a controller for the drive system and a filter that incrementally estimates the direction of motion. Rather than knowing the planned trajectory or estimating the precise joint configuration of the mechanism, the robot applies compliant forces and dynamically adjusts its motion based on the observed end-effector velocity. The system is able to explore and adaptively follow constrained paths even when faced with sharp turns or uncertainties.

Karayiannidis et al. treats the task as a model-free, adaptive interaction problem [3]. Their controller simultaneously estimates whether the joint is prismatic or revolute and learns the corresponding motion axis, relying solely on wrist force/torque signals. A force-velocity scheme drives the end-effector so that tangential motion accomplishes the task, while forces in the constrained directions are suppressed.

The study of mobile manipulators, which combine robotic arms with wheeled platforms, is key to extending the adaptive control method to mobile robots. Several works focus on control frameworks that tightly integrate the base and manipulator dynamics. In [5], a quadratic-programming framework treats the wheels and arm as a single kinematic chain, balancing end-effector tracking with joint-limit, velocity, and non-holonomic constraints to yield fluid "whole-body" motions. The paper [6] proposes a slack-based QP that maximizes manipulability and steers the robot away from singularities without relying on redundancy alone. Classical insights are provided by [7], which unifies the kinematics of wheeled platforms and serial arms and adapts reduced gradient schemes for redundancy resolution under non-holonomic restrictions. The paper [8] further quantifies how adding a mobile base reshapes the manipulability ellipsoid and guides criterion-driven path generation. A task-priority perspective is taken in [3], where a hybrid switching law guarantees bounded joint speeds, detours around singular configurations, and preserves the primary end-effector objective.

Complementing these controller designs, a number of works address the broader integration and real-world applicability of mobile manipulation systems. For instance, [9] introduces an architecture that performs pick-and-place actions while the base moves towards the next goal, using behavior trees and a holistic velocity controller to exceed 99 % success in real runs. When grasps fail, the reactive base placement strategy in [10] prolongs the time the object stays within reach, recovering without sacrificing the speed gains of on-the-move manipulation. Obstacle-rich environments are considered in [11], which augments the holistic QP with arm and base avoidance constraints and integrates a short-horizon A* planner to allow the robot to weave past static and dynamic hazards while manipulating. Finally, the survey in [12] highlights how compliant, impedance-based behaviors are increasingly fused with such reactive schemes to handle force-critical tasks, underscoring the trend towards fast, adaptable, and principled control solutions for mobile manipulation.

To address these challenges, this thesis proposes a unified control and estimation framework that enables a differential-drive mobile manipulator

to robustly interact with 1-DoF mechanisms of unknown kinematic structure. Building upon the adaptive control approach introduced by Karayiannidis et al. [3], we formulate the following key research objectives:

- **Objective 1**: How can a differential-drive mobile manipulator autonomously navigate to suitable pre-interaction poses and achieve coordinated whole-body motion that ensures safe, smooth, and efficient execution of manipulation tasks with 1-DoF mechanisms?
- **Objective 2**: How can an online estimation framework be developed to reliably learn the kinematic properties—such as motion direction and curvature—of unknown 1-DoF mechanisms during interaction, and how can this estimation enhance task execution?

Our contribution is to embed mobility into the interaction process, turning base motion into an integral part of the manipulation strategy. All objectives are evaluated through extensive simulations and real-world experiments across a diverse set of 1-DoF mechanism types.

2

System and Problem Description

The complete door-opening mission executed by the mobile manipulator can be divided as: (i) autonomously navigating from an arbitrary start pose to a pre-grasp way point in front of the door handle, (ii) opening the door by grasping the handle and coordinating the motions of the differential-drive base and 6-DoF arm, and (iii) continuously estimating the door's unknown kinematic parameters. During the opening phase, we employ the adaptive velocity controller proposed in [3]. We now define the problem in detail, following the three functional blocks—navigation, whole body control, and online estimator.

2.1 Frames and Notation

We introduce the following notation for coordinate frames:

- The **world/inertial frame** is denoted by $\{W\}$. It serves as the fixed inertial reference frame for all transformations and is typically aligned with the ground or the environment.
- The **mobile base frame** is denoted by $\{MB\}$. This frame is rigidly attached to the mobile base, located at the geometric center between the drive wheels.
- The **arm base frame** is denoted by $\{AB\}$. It is attached to the mounting point of the robotic arm on the mobile base.
- The **end-effector frame** is denoted by $\{E\}$. This frame is attached to the tool center point (TCP) of the manipulator.

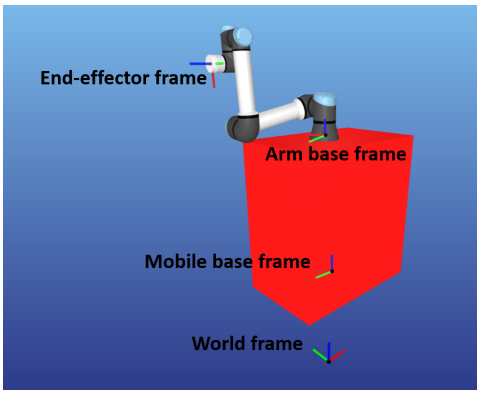


Figure 2.1 Four different frames

Figure 2.1 shows the frames. The transformation from one frame to another is described using the standard notation:

- ${}^{i}\cdot$ represents a variable expressed in frame $\{i\}$. For example, ${}^{i}\mathbf{t} \in \mathbb{R}^3$ is a 3D position vector and ${}^{i}\mathbf{R} \in SO(3)$ is a 3D rotation matrix described in frame $\{i\}$.
- ${}^{j}\mathbf{t}_i \in \mathbb{R}^3$ is the transition vector from frame $\{j\}$ to frame $\{i\}$, expressed in frame $\{j\}$.
- ${}^{j}\mathbf{R}_i \in SO(3)$ is the rotation matrix that transforms vectors from frame $\{i\}$ to frame $\{j\}$.

For example, given a pose described in the arm base frame by ${}^{AB}\mathbf{p}$ and ${}^{AB}\mathbf{R}$, the same pose can be represented in the world frame as follows:

$$ {}^{W}\mathbf{p} = {}^{W}\mathbf{R}_{AB} \cdot {}^{AB}\mathbf{p} + {}^{W}\mathbf{t}_{AB} \tag{2.1} $$

$$ {}^{W}\mathbf{R} = {}^{W}\mathbf{R}_{AB} \cdot {}^{AB}\mathbf{R} \tag{2.2} $$

We introduce some other notation:

- $[\cdot]_\times$ denotes the skew-symmetric matrix operator. For a vector $\omega = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^T \in \mathbb{R}^3$, the corresponding skew-symmetric matrix $[\omega]_\times$ is given by:

$$ [\omega]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.3} $$

- $\mathbf{P}(\cdot)$ denotes the projection matrix operator. For a unit vector $\mathbf{x} \in \mathbb{R}^3$, the corresponding skew-symmetric matrix $\mathbf{P}(\mathbf{x})$ is given as follows:

$$\mathbf{P}(\mathbf{x}) = \mathbf{I}_3 - \mathbf{x}\mathbf{x}^T \tag{2.4}$$

- Throughout this paper, the spatial twist is consistently represented as $\mathbf{V} = \begin{bmatrix} \boldsymbol{\omega}^T & \mathbf{v}^T \end{bmatrix}^T$, with angular velocity preceding linear velocity.

2.2 Navigation

Inverse kinematics (IK) is widely used in robotic manipulators where the mapping between control inputs and task-space motion is well-defined and fully actuated [12]. Classical inverse kinematics assumes a linear, differentiable relationship between joint velocities and end-effector motion.

A differential-drive mobile base differs fundamentally from a normal manipulator because of its underactuated and nonholonomic nature. The robot configuration is typically described by the state vector $q = (x, y, \theta)$, but only two control inputs—the left and right wheel velocities or, equivalently, the turning and forward velocities (ω, v)—can be commanded. The constraint leads to the base being unable to move sideways (along its local y-axis). Lateral movements must be achieved via a sequence of forward motions and rotations. This violates a fundamental assumption in inverse kinematics, which requires a full-rank Jacobian that maps task space velocities to control inputs.

Rather than relying on direct IK solutions, differential-drive bases require nonlinear feedback controllers to ensure smooth convergence to target poses. These controllers account for both position and orientation, overcoming the limitations of linear IK solvers.

2.3 Whole Body Control (WBC)

As mentioned above, IK implicitly assumes that every task-space DoF is backed by an independent control input. In a mobile manipulator, this premise is broken in two ways: (i) the differential-drive base is underactuated and non-holonomic, and (ii) the arm joints, together with the wheels, provide more actuators than the task actually requires. WBC therefore reforms the control objective from "solve a square IK" to "coordinate a redundant but partly uncontrollable system so that primary tasks are met and leftover freedom is steered towards optimal secondary goals." This

perspective is consistent with findings in humanoid robotics [13], where opening a heavy swing door requires the robot to exploit whole-body momentum transfer and coordinated arm-base motion. Static manipulation with the arm alone was insufficient; instead, stability and task feasibility could only be ensured through carefully planned whole-body actions that balance locomotion, impact generation, and post-impact support.

Primary Goal: Task Feasibility under Under-Actuation

The combined configuration vector can be represented as follows:

$$q = \begin{bmatrix} x_r & y_r & \theta_r & \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 \end{bmatrix}^T \tag{2.5}$$

where x_r, y_r, θ_r denote the planar position and orientation of the differential-drive base, and $\theta_1, \ldots, \theta_6$ are the joint angles of the 6-DoF manipulator arm. The configuration vector thus contains **9** DoF, but only **8** independent control inputs are available, consisting of 2 inputs from the mobile base—the forward velocity v_b and the angular velocity ω_b—together with 6 inputs $\dot{\theta}_1, \ldots, \dot{\theta}_6$ from the manipulator joints.

To capture the coupled kinematics of the 6-DoF manipulator and the differential-drive base, the end-effector twist in frame $\{W\}$, expressed as ${}^W\mathbf{V}_{ee} \triangleq \begin{bmatrix} {}^W\boldsymbol{\omega}_{ee}^T & {}^W\mathbf{v}_{ee}^T \end{bmatrix}^T \in \mathbb{R}^6$ (${}^W\boldsymbol{\omega}_{ee} \in \mathbb{R}^3$ and ${}^W\mathbf{v}_{ee} \in \mathbb{R}^3$ denote the rotational and translational velocity respectively), is obtained from joint space through the following mapping relationship:

$${}^W\mathbf{V}_{ee} = \mathbf{J}^* \begin{bmatrix} \dot{\mathbf{q}} \\ {}^W\mathbf{u} \end{bmatrix} \tag{2.6}$$

where $\dot{\mathbf{q}} \in \mathbb{R}^6$ represents the joint velocities of the 6-DOF robotic arm, ${}^W\mathbf{u} \triangleq \begin{bmatrix} {}^W\omega_b & {}^W v_b \end{bmatrix}^T \in \mathbb{R}^2$ (${}^W\omega_b \in \mathbb{R}$ and ${}^W v_b \in \mathbb{R}$ denote the turning velocity and forward velocity of the mobile base respectively), and $\mathbf{J}^* \in \mathbb{R}^{6\times 8}$ is the extended Jacobian matrix.

Then we can implement inverse kinematics as follows:

$$\begin{bmatrix} \dot{\mathbf{q}} \\ {}^W\mathbf{u} \end{bmatrix} = \mathbf{J}^+ \, {}^W\mathbf{V}_{ee} \tag{2.7}$$

with $\mathbf{J}^+ = \mathbf{J}^{*T}[\mathbf{J}^*\mathbf{J}^{*T}]^{-1}$ being the pseudo-inverse.

Redundancy: Using Extra Freedom

When the number of controllable degrees of freedom (DoFs) exceeds the number of task-space constraints, the robotic system is kinematically redundant [14]. Mathematically, because $\mathbf{J}^* \in \mathbb{R}^{6\times 8}$ in (2.6) is tall, infinitely

many $\dot{\mathbf{q}}$ and ${}^{W}\mathbf{u}$ satisfy the same ${}^{W}\mathbf{V}_{ee}$. Redundancy is not a burden; it is latent freedom that can be organised so the robot simultaneously fulfils the primary task and optimises additional criteria (e.g., joint-limit avoidance, manipulability, collision avoidance) without disturbing the primary task. In

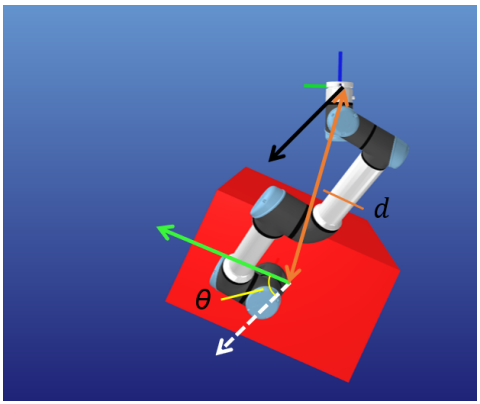


Figure 2.2 The definitions of d and θ. The green and black arrows represent the velocity directions of the base and the end-effector, respectively. The white arrow is parallel to the black arrow, indicating the intended direction of end-effector motion.

the proposed framework, two key secondary objectives are considered to ensure safe and feasible motion coordination: maintaining a safe distance between the base and the end-effector, and aligning the base's heading with the end-effector's motion direction. The two quantities that need to be controlled are shown in Figure 2.2.

(i) Maintaining a Safe Base-to-End-Effector Distance: To avoid self-collision and maintain spatial coordination, the Euclidean distance between the base and the end-effector is defined as follows:

$$d = \sqrt{(x_e - x_b)^2 + (y_e - y_b)^2} \tag{2.8}$$

where (x_e, y_e) and (x_b, y_b) denote the positions of the end-effector and the base in $\{W\}$, respectively. The objective is to ensure that d converges to the pre-defined target value d_{target}.

(ii) Aligning the Base's Heading with the End-effector's Motion Direction: By aligning the base orientation with the end-effector velocity, the system avoids excessive arm compensation and enhances overall task feasibility, taking into account the underactuated nature of the differential-drive base.

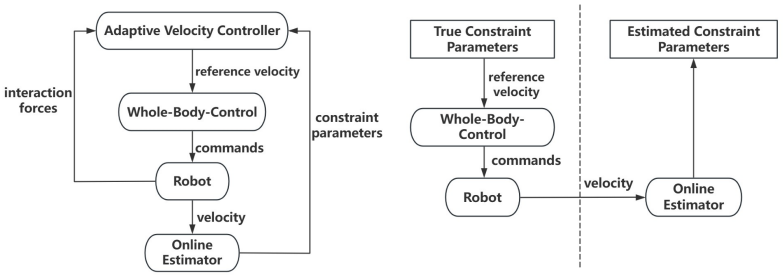
Since the differential-drive base can only move along its forward direction, its heading always coincides with the direction of its velocity. Let $\mathbf{v}_e = [\dot{x}_e, \dot{y}_e]^T$ be the end-effector's planar velocity, and $\mathbf{v}_b = [\dot{x}_b, \dot{y}_b]^T$ the forward motion components of the base. The misalignment angle is computed as follows:

$$\theta = \arccos\left(\frac{\mathbf{v}_e \cdot \mathbf{v}_b}{\|\mathbf{v}_e\|\|\mathbf{v}_b\|}\right) \tag{2.9}$$

This task-priority strategy guarantees that the robot achieves its main objectives while utilizing redundancy to satisfy additional constraints that improve safety and motion feasibility.

2.4 Online Estimator

While whole-body control solves the problem of coordinating an underactuated mobile base and redundant manipulators, it does so given a reference velocity of the end-effector. However, the reference velocity is given by an adaptive velocity controller [3], which assumes that the constraint parameters of the mechanism are known. To address this problem, we introduce an online estimator that continuously learns these constraints during the task by observing the velocity magnitude and direction over time.



(a) Force-Feedback Architecture **(b)** Decoupled Architecture

Figure 2.3 Comparison between force-feedback-based and decoupled control architectures.

The estimator computes the constraint parameters, e.g., motion axis and curvature, based on past velocity information of the end-effector. These estimated parameters are then passed to the adaptive velocity controller. By combining them with the interaction forces measured at the end-effector, the adaptive velocity controller gives a reference velocity, which

ensures the motion of the end-effector in the allowed direction by suppressing the force component in the constrained direction. Ideally, the estimated parameters converge to the true values under the assumption of continuous execution and valid grasping.

However, this thesis does not incorporate any form of force feedback. Therefore, to verify the performance of the estimator, we completely decouple the online estimator from the controller. We compute the reference velocity using the real (assumed) kinematic constraints and feed it directly into the whole-body control module. In this setup, the estimator runs in parallel with the controller. We evaluate its performance by observing whether it can successfully converge to known kinematic constraints, thereby verifying its accuracy and robustness when out of the force feedback control loop. Figure 2.3 shows the difference between force-feedback-based and decoupled control architectures.

3

Navigation

To enable the mobile base to autonomously reach a desired pose in front of the door, we adopt a nonlinear feedback controller tailored for differential-drive robots [12]. This approach respects the underactuated and non-holonomic nature of the platform, and computes a velocity command that ensures convergence in both position and orientation.

Control Law Formulation

The configuration of the mobile base is described in the world frame by the vector

$$^{W}q_{base} = (x_r, y_r, \theta_r) \tag{3.1}$$

where x_r, y_r denotes the base's Cartesian coordinates in the world frame, and θ_r is its heading angle. The goal is to design a control law that drives the base to a target pose (x_t, y_t, θ_t) by generating appropriate forward and turning velocities.

To begin with, we define a polar coordinate transformation centered on the target pose, which can easily represent relative position and orientation. As is shown in Figure 3.1:

- ρ is the Euclidean distance between the robot and the goal,
- γ is the angular deviation between the direction to the target and the robot's forward-facing direction,
- δ is the angular deviation between the direction to the target and the target's forward-facing direction.

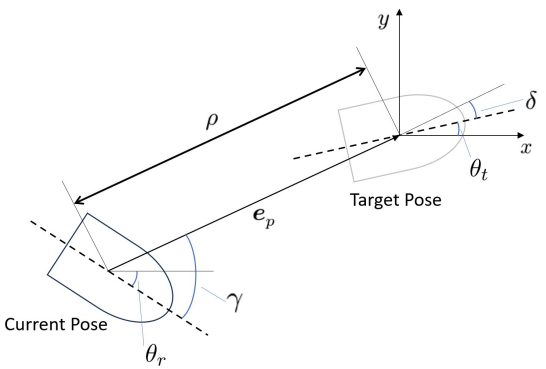


Figure 3.1 Definition of polar coordinates for the differential drive base

These parameters are computed as follows:

$$\rho = \sqrt{(x_r - x_t)^2 + (y_r - y_t)^2} \tag{3.2}$$

$$\gamma = \arctan 2(y_r - y_t, x_r - x_t) - \theta_r \tag{3.3}$$

$$\delta = \gamma + \theta_r - \theta_t \tag{3.4}$$

The control law is given as follows [12]:

$$v = k_1 \rho \cos\gamma \tag{3.5}$$

$$\omega = k_2 \gamma + k_1 \left(\frac{\sin\gamma \cos\gamma}{\gamma} \right) (\gamma + k_3 \delta) \tag{3.6}$$

where:

- v is the commanded forward velocity,
- ω is the commanded turning velocity,
- k_1, k_2, k_3 are positive control gains that regulate convergence rate and stability.

The forward speed command (3.5) can be interpreted as the projection of the translational error ρ onto the robot's forward-facing direction.

When the heading error γ is small, $\cos\gamma \approx 1$ and the robot advances almost at the full rate $k_1\rho$, maximizing navigation progress. On the contrary, as $|\gamma|$ grows, the cosine term decreases, limiting forward motion and causing the base to rotate first before reaching the destination.

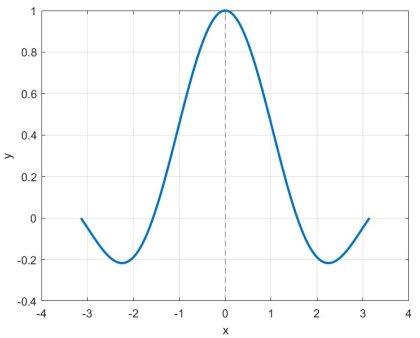


Figure 3.2 Function graph of $y = \frac{\sin x \cos x}{x}$ for $x \in (-\pi, \pi)$.

This design prevents the robot from turning too early or too sharply near the goal and consequently missing the intended final orientation.

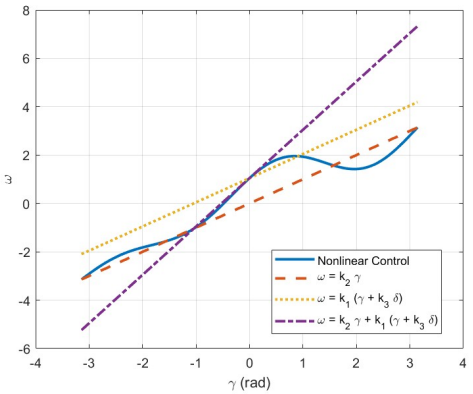


Figure 3.3 Comparison of turning velocity ω under nonlinear and linear controllers with fixed $\delta = \pi/6$.

The turning speed command (3.6) consists of two parts. The first term, $k_2\gamma$ is a standard proportional correction. The second term introduces a smooth gain scheduler through the term $\frac{\sin\gamma\cos\gamma}{\gamma}$. As is shown in Figure 3.2, when γ is relatively large, $\left|\frac{\sin\gamma\cos\gamma}{\gamma}\right|$ tends to be small, which helps to suppress overcorrections in case of large errors. From Figure 3.3 we can see that, at this time, $k_2\gamma$ dominates the control of ω.

As γ decreases from a large value to near zero, $\frac{\sin\gamma\cos\gamma}{\gamma}$ approaches 1, in which case the contribution of the term $(\gamma + k_3\delta)$ that combines the heading error with the final pose orientation error, allowing the robot to gradually adjust to the desired heading as it approaches the target, becomes relatively more significant.

The design of the controller balances responsiveness and smoothness, prevents overshoot, and ensures convergence of position and orientation targets, thus forming the basis for the first phase of the door-opening task.

4

Whole Body Control

4.1 Extended Jacobian

Here we describe how to compute the $\mathbf{J}^*$ in (2.6). ${}^W\mathbf{V}_{ee}$ can be decomposed into the contributions of the manipulator joints and the mobile base. Mathematically, we can rewrite (2.6) into:

$$ {}^W\mathbf{V}_{ee} = \begin{bmatrix} \mathbf{J}_1^* & \mathbf{J}_2^* \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}} \\ {}_W\mathbf{u} \end{bmatrix} \longrightarrow {}^W\mathbf{V}_{ee} = \mathbf{J}_1^*\dot{\mathbf{q}} + \mathbf{J}_2^{*W}\mathbf{u} = {}^W\mathbf{V}'_{ee} + {}^W\mathbf{V}_b \quad (4.1) $$

Specifically, $\mathbf{J}_1^*$ maps the joint velocity vector $\dot{\mathbf{q}} \in \mathbb{R}^6$ to the end-effector twist, while $\mathbf{J}_2^*$ maps the base control input ${}^W\mathbf{u} \in \mathbb{R}^2$ (i.e., turning and forward velocities) into the same twist space.

It should be noted that the decomposition ${}^W\mathbf{V}_{ee} = {}^W\mathbf{V}'_{ee} + {}^W\mathbf{V}_b$ should not be interpreted as an accumulation of relative motion. Instead, it represents a linear superposition of two twist components, both expressed in the same inertial frame $\{W\}$. The term ${}^W\mathbf{V}'_{ee} = \mathbf{J}_1^*\dot{\mathbf{q}}$ corresponds to the twist from the arm's joint motion, while ${}^W\mathbf{V}_b = \mathbf{J}_2^{*W}\mathbf{u}$ accounts for the contribution from the base motion. Since both components are transformed into the inertial frame, their sum yields the true end-effector twist in $\{W\}$.

Computation of $\mathbf{J}_1^*$

In the manipulator base frame, the standard geometric Jacobian is

$$ {}^{AB}\mathbf{V}_{ee} = \mathbf{J}_m\dot{\mathbf{q}} \quad (4.2) $$

where $\mathbf{J}_m$ represents the manipulator Jacobian defined in the local frame of the manipulator base. Because $\mathbf{J}_m$ only maps the joint velocities $\dot{\mathbf{q}}$ to the end-effector twist, $\mathbf{J}_m$ depends only on the manipulator's kinematics, which do not change unless the joint configuration $\mathbf{q}$ changes.

In fact, ${}^{AB}\mathbf{V}_{ee}$ in (4.2) is exactly the same twist as ${}^{W}\mathbf{V}'_{ee}$ in (4.1), which accounts for the contribution of the arm joints to the end-effector twist, but just expressed in different frames. Because the conventional manipulator Jacobian $\mathbf{J}_m$ is derived in the arm base frame $\{AB\}$, a change of coordinates is required. Thus, the mapping relationship between $\mathbf{J}_m$ and $\mathbf{J}_1^*$ is established through the transformation of ${}^{AB}\mathbf{V}_{ee}$ to ${}^{W}\mathbf{V}'_{ee}$.

The adjoint transformation matrix is often used to transform twists in different reference frames [14]. Let the homogeneous transformation matrix from $\{AB\}$ (arm base) to the world frame $\{W\}$ be

$$ {}^{W}T_{AB} = \begin{bmatrix} {}^{W}R_{AB} & {}^{W}\mathbf{t}_{AB} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{4.3} $$

However, we can not get ${}^{W}R_{AB}$ and ${}^{W}\mathbf{t}_{AB}$ directly. Another transformation is needed:

$$ {}^{W}T_{AB} = {}^{W}T_{MB} \cdot {}^{MB}T_{AB} \tag{4.4} $$

Then we get:

$$ {}^{W}R_{AB} = {}^{W}R_{MB} \cdot {}^{MB}R_{AB}, \quad {}^{W}\mathbf{t}_{AB} = {}^{W}R_{MB} \cdot {}^{MB}\mathbf{t}_{AB} + {}^{W}\mathbf{t}_{MB} \tag{4.5} $$

where

- ${}^{W}R_{MB}$ and ${}^{W}\mathbf{t}_{MB}$ represent the mobile base's current pose, typically derived from a localization system.
- ${}^{MB}R_{AB}$ and ${}^{MB}\mathbf{t}_{AB}$ define the static transformation from the arm base frame $\{AB\}$ to the mobile base frame $\{MB\}$, which is determined by the mechanical configuration of the mobile manipulator.

The corresponding adjoint matrix reads:

$$ \mathrm{Ad}_{({}^{W}T_{AB})} = \begin{bmatrix} {}^{W}R_{AB} & \mathbf{0}_{3\times 3} \\ [{}^{W}\mathbf{p}_{AB}]_{\times}\, {}^{W}R_{AB} & {}^{W}R_{AB} \end{bmatrix} \tag{4.6} $$

The twist of the end-effector generated by joint motion alone can now be mapped into $\{W\}$ as follows:

$$ {}^{W}\mathbf{V}'_{ee} = \mathrm{Ad}_{({}^{W}T_{AB})}\, {}^{AB}\mathbf{V}_{ee} \iff {}^{W}\mathbf{V}'_{ee} = \mathrm{Ad}_{({}^{W}T_{AB})}\, \mathbf{J}_m \dot{\mathbf{q}} \tag{4.7} $$

Hence

$$ \mathbf{J}_1^* = \mathrm{Ad}_{({}^{W}T_{AB})}\, \mathbf{J}_m \tag{4.8} $$

Concretely, we can view the adjoint matrix as

- It applies a 1:1 linear transformation (i.e., rotation) to the angular velocity ω.

- It not only rotates the linear velocity, but also adds a coupling term determined by the position vector $\mathbf{p}$ and the ω after rotation.

It may sound a bit strange that the contribution of linear velocity is affected by angular velocity. While the angular velocity of a rigid body is invariant with respect to the point of observation, the linear velocity depends on the chosen reference point as a result of the term $\omega \times \mathbf{r}$, which captures how rotational motion induces different tangential speeds at different positions. This position dependence is precisely why the adjoint matrix includes a coupling term in the linear velocity component.

Computation of $\mathbf{J}_2^*$

In this thesis, we directly command forward and turning velocities, so it is quite simple to model the control system. Figure 4.1 shows a differential drive robot.

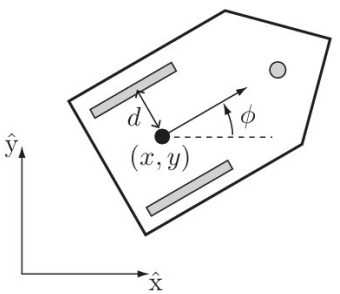


Figure 4.1 A differential drive robot [14]

Select the point in the middle of the two wheels as the reference point. We have:

$$ {}^{W}\mathbf{V}_b = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & \cos\theta_r \\ 0 & \sin\theta_r \\ 0 & 0 \end{bmatrix} \begin{bmatrix} {}^{W}\omega_b \\ {}^{W}v_b \end{bmatrix} \tag{4.9} $$

Hence

$$\mathbf{J}_2^* = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & \cos\theta_r \\ 0 & \sin\theta_r \\ 0 & 0 \end{bmatrix} \tag{4.10}$$

Summary

The extended Jacobian $\mathbf{J}^*$ maps the generalized velocities (including both manipulator joint velocities and base velocities) to the end-effector twist in the world frame. It is defined as follows:

$$\mathbf{J}^* = \begin{bmatrix} \mathbf{J}_1^* & \mathbf{J}_2^* \end{bmatrix} \tag{4.11}$$

where:

- $\mathbf{J}_1^* \in \mathbb{R}^{6\times 6}$ is the contribution of the manipulator arm,
- $\mathbf{J}_2^* \in \mathbb{R}^{6\times 2}$ is the contribution of the differential-drive mobile base.

Manipulator contribution:

$$\mathbf{J}_1^* = \begin{bmatrix} {}^W R_{AB} & \mathbf{0}_{3\times 3} \\ [{}^W\mathbf{p}_{AB}]_\times {}^W R_{AB} & {}^W R_{AB} \end{bmatrix} \mathbf{J}_m \tag{4.12}$$

where:

- $\mathbf{J}_m$: geometric Jacobian of the manipulator (expressed in the arm base frame $\{AB\}$),
- ${}^W R_{AB}$: rotation from frame $\{AB\}$ (manipulator base) to world frame $\{W\}$,
- ${}^W\mathbf{p}_{AB}$: position of frame $\{AB\}$ in frame $\{W\}$,
- $[\cdot]_\times$: skew-symmetric matrix operator (used to compute cross products).

Base contribution:

$$\mathbf{J}_2^* = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & \cos\theta_r \\ 0 & \sin\theta_r \\ 0 & 0 \end{bmatrix} \tag{4.13}$$

where:

- θ_r: heading angle of the differential-drive base (orientation around the vertical axis).

4.2 Optimization

Primary Task-Level Kinematic Optimization

The primary task is to realize the desired spatial velocity $\mathbf{v}_{\text{ref}} \in \mathbb{R}^6$ of the end-effector. In a mobile robot with redundant degrees of freedom, the same end-effector velocity can be achieved by an infinite number of joint speed/angle combinations. Since the robotic arm has more accurate positioning and lower energy consumption, the control strategy prioritizes achieving the task through the manipulator, reserving base movement only when necessary.

We adopt a weighted damped pseudo-inverse that produces the velocity command $\dot{\mathbf{q}}_{\text{task}} \in \mathbb{R}^8$ as the least-squares solution of the following constrained optimization problem [15]:

$$\min_{\dot{\mathbf{q}}_{\text{task}}} \left\| \mathbf{W}^{\frac{1}{2}} \dot{\mathbf{q}}_{\text{task}} \right\|^2 \quad \text{s. t.} \quad \mathbf{J}^* \dot{\mathbf{q}}_{\text{task}} = \mathbf{v}_{\text{ref}} \tag{4.14}$$

where $\mathbf{W} \in \mathbb{R}^{8\times 8}$ is a symmetric positive-definite weight matrix.

This problem seeks the joint velocity vector $\dot{\mathbf{q}}$ that achieves the exact desired end-effector velocity while minimizing the weighted 2-norm $\|\mathbf{W}^{\frac{1}{2}} \dot{\mathbf{q}}_{\text{task}}\|^2 = \dot{\mathbf{q}}_{\text{task}}^T \mathbf{W} \dot{\mathbf{q}}_{\text{task}}$. The solution favors configurations where joints or actuators with smaller weights move more than those with higher weights, thereby optimizing control effort under specified priorities.

The corresponding solution is given by the weighted damped pseudo-inverse [16]:

$$\mathbf{J}_W^+ = \mathbf{W}^{-1} \mathbf{J}^{*T} \left(\mathbf{J}^* \mathbf{W}^{-1} \mathbf{J}^{*T} + \lambda I_6 \right)^{-1} \tag{4.15}$$

where λ is an adaptive damping factor that improves robustness near singular configurations. The commanded joint velocity is then

$$\dot{\mathbf{q}}_{\text{task}} = \mathbf{J}_W^+ \mathbf{v}_{\text{ref}} \tag{4.16}$$

Choice of the Weight Matrix W We partition the 8 controllable DoFs into the 6 DoFs of the manipulator arm and the 2 DoFs of the mobile base. Accordingly,

$$\mathbf{W} = \text{diag}\big(w_a I_6, w_b I_2\big), \qquad w_b \gg w_a > 0 \tag{4.17}$$

Because (4.15) minimizes the weighted 2-norm $\|W^{\frac{1}{2}}\dot{\mathbf{q}}\|$, large coefficients w_b penalize base motions far more than arm motions. Therefore, the robot will always prioritize mobilizing the robotic arm to complete the task.

Adaptive Damping Factor λ When it is close to the singular configuration $\det(\mathbf{J}\mathbf{J}^T) \to 0$, the undamped pseudoinverse will produce excessively high joint velocities. We approach this with an adaptive Tikhonov term

$$\mu = \sqrt{\det\left(\mathbf{J}\mathbf{J}^T\right)} \tag{4.18}$$

$$\lambda = \frac{\lambda_0}{\mu + \varepsilon} \tag{4.19}$$

where μ is Yoshikawa's manipulability index [8], $\lambda_0 > 0$ a positive gain parameter and $\varepsilon > 0$ prevents division by zero. As the singularity is approached, μ decreases, leading to λ rising smoothly, increasing the diagonal of (4.15) and modulating the joint velocities in (4.16). This preserves numerical stability without sacrificing responsiveness away from the singularity region.

Equations (4.14) – (4.19) achieve a weighted damped pseudo-inverse that prioritizes arm motion over base movement and ensures stability near singularities with an adaptive damping strategy.

Null-Space Optimization

In the previous section, we minimized the base motion while accomplishing the primary task. However, beyond fulfilling the main objective, it is vital to fully exploit the redundant degrees of freedom. By doing so, we can improve posture, avoid collisions, or enhance manipulability. We refer to these optimization goals as secondary tasks. These secondary tasks must be accomplished without interfering with the motion of the end-effector.

Null Space of the Jacobian The null space of the Jacobian matrix $\mathbf{J}^*$ is a fundamental concept in the control of redundant robotic systems. It is defined as the set of all joint velocity vectors that do not produce any end-effector motion [17], i.e.,

$$\mathbf{N}(\mathbf{J}^*) = \{\dot{\mathbf{q}}_{\text{null}} \in \mathbb{R}^n \mid \mathbf{J}^*\dot{\mathbf{q}}_{\text{null}} = \mathbf{0}\} \tag{4.20}$$

Geometrically, the null space can be interpreted as the directions in the joint velocity space that are "invisible" to the task space; in other words,

motion along these directions does not affect the end-effector velocity. This property is particularly useful in the context of kinematically redundant systems, such as mobile manipulators, where the number of control inputs exceeds the dimension of the task.

In practical terms, any joint velocity vector $\dot{\mathbf{q}}_{\text{null}}$ that lies in $\mathbf{N}(\mathbf{J}^*)$ can be exploited to achieve secondary objectives without interfering with the primary task execution. The general solution to the inverse kinematics problem in the presence of redundancy takes the following form [17]:

$$\dot{\mathbf{q}}^* = \underbrace{\mathbf{J}_W^{*+}\mathbf{v}_{\text{ref}}}_{\dot{\mathbf{q}}_{\text{task}}} + \underbrace{\left(\mathbf{I} - \mathbf{J}^{*+}\mathbf{J}^*\right)\mathbf{z}}_{\dot{\mathbf{q}}_{\text{null}}} \tag{4.21}$$

where $\dot{\mathbf{q}}^* \in \mathbb{R}^8$ is the overall velocity command sent to the robot and the vector $\mathbf{z} \in \mathbb{R}^8$ is an arbitrary design parameter that can be selected to drive the system toward desirable configurations based on secondary criteria. The projection matrix

$$\mathbf{N}(\mathbf{q}) \triangleq \mathbf{I} - \mathbf{J}^{*+}\mathbf{J}^* \tag{4.22}$$

acts as a null-space projector that filters out the components of any velocity vector that would interfere with the primary task.

This formulation ensures that the contribution of the secondary objective is orthogonal to the task Jacobian, thereby preserving the desired end-effector behavior.

Through (4.21), the coordinated motion between the base and the manipulator is achieved: the primary task is maintained by the pseudo-inverse term, while redundant motions (such as optimizing the base motion or the arm posture) can be injected through the null space component. This approach achieves the optimization of the secondary task during the execution of the primary task.

Choice of the secondary velocity z A commonly used strategy is the projected-gradient method, which sets [18]:

$$\mathbf{z} = -\alpha \nabla_{\mathbf{q}} f(\mathbf{q}) \tag{4.23}$$

where $f(\mathbf{q})$ is the cost function that encodes the secondary objective and $\alpha > 0$ is a gain parameter. (4.23) is motivated by the fact that the gradient $\nabla_{\mathbf{q}} f(\mathbf{q})$ points in the direction of rising cost function. Thus, the negative gradient $-\nabla_{\mathbf{q}} f(\mathbf{q})$ provides a descent direction that can be used to minimize $f(\mathbf{q})$.

$\mathbf{z}_1$: keeping a safe distance between the arm and base. The cost function in (4.23) is defined as follows:

$$f(\mathbf{q}) = \frac{1}{2}(d - d_{target})^2 \tag{4.24}$$

where the term d denotes the Euclidean distance between the mobile base and the end-effector measured in (2.8), while d_{target} is the desired reference distance. According to chain rule

$$\nabla_{\mathbf{q}} f(\mathbf{q}) = (d - d_{target}) \frac{\partial d}{\partial \mathbf{q}} \tag{4.25}$$

since $d = d(\mathbf{q}(t))$ depends on the generalized coordinates, its time derivative satisfies

$$\dot{d} = \frac{\partial d}{\partial \mathbf{q}} \dot{\mathbf{q}} \tag{4.26}$$

which implies

$$\frac{\partial d}{\partial \mathbf{q}} = \frac{\partial \dot{d}}{\partial \dot{\mathbf{q}}} \tag{4.27}$$

Substituting (4.27) into (4.25) gives:

$$\nabla_{\mathbf{q}} f(\mathbf{q}) = (d - d_{target}) \frac{\partial \dot{d}}{\partial \dot{\mathbf{q}}} \tag{4.28}$$

Differentiating d with respect to time gives:

$$\dot{d} \;=\; \frac{(x_e - x_b)(\dot{x}_e - \dot{x}_b) + (y_e - y_b)(\dot{y}_e - \dot{y}_b)}{d} \tag{4.29}$$

which couples the linear velocities of the end-effector, $\dot{x}_e, \dot{y}_e$, and of the mobile base, $\dot{x}_b, \dot{y}_b$. Introducing the shorthand $dx = x_e - x_b$ and $dy = y_e - y_b$ we rewrite (4.29) as follows:

$$\dot{d} \;=\; \frac{dx(\dot{x}_e - \dot{x}_b) + dy(\dot{y}_e - \dot{y}_b)}{d} \tag{4.30}$$

The Cartesian velocities of the end-effector are obtained from the first two rows of the whole-body Jacobian,

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \end{bmatrix} \;=\; \begin{bmatrix} \mathbf{J}_x^* \\ \mathbf{J}_y^* \end{bmatrix} \dot{\mathbf{q}} \tag{4.31}$$

where $\dot{\mathbf{q}} \in \mathbb{R}^8$ concatenates the joint-rate vector of the six-DoF arm and the planar twist ${}^W\mathbf{u} = [{}^W\omega_b \; {}^W v_b]^T$ of the differential drive.

The Cartesian velocities of the base can be obtained from (4.9):

$$\begin{bmatrix} \dot{x}_b \\ \dot{y}_b \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & \cos\theta_r \\ 0 & 0 & \cdots & 0 & \sin\theta_r \end{bmatrix} \dot{\mathbf{q}} \tag{4.32}$$

Substituting (4.31) – (4.32) into (4.30) yields

$$\dot{d} = \frac{1}{d} \begin{bmatrix} dx & dy \end{bmatrix} \left(\begin{bmatrix} \mathbf{J}_x^* \\ \mathbf{J}_y^* \end{bmatrix} - \begin{bmatrix} 0 & \cdots & 0 & \cos\theta_r \\ 0 & \cdots & 0 & \sin\theta_r \end{bmatrix} \right) \dot{\mathbf{q}} \triangleq \mathbf{J}_d \dot{\mathbf{q}} \tag{4.33}$$

where $\mathbf{J}_d = \frac{\partial d}{\partial \mathbf{q}} \in \mathbb{R}^{1\times 8}$ is the gradient of the scalar distance with respect to the generalized velocities. Then we have:

$$\nabla_{\mathbf{q}} f(\mathbf{q}) = (d - d_{target}) \mathbf{J}_d^T \tag{4.34}$$

Substituting (4.34) into (4.23) gives the secondary velocity command:

$$\mathbf{z}_1 = -\alpha_1 (d - d_{\text{target}}) \mathbf{J}_d^T \tag{4.35}$$

where $\mathbf{J}_d = \frac{1}{d} \begin{bmatrix} dx & dy \end{bmatrix} \left(\begin{bmatrix} \mathbf{J}_x^* \\ \mathbf{J}_y^* \end{bmatrix} - \begin{bmatrix} 0 & \cdots & 0 & \cos\theta_r \\ 0 & \cdots & 0 & \sin\theta_r \end{bmatrix} \right)$ and the positive gain α_1 tunes the convergence speed. The term (4.35) reduces the arm-base distance whenever $d > d_{\text{target}}$ and increases it whenever $d < d_{\text{target}}$.

$\mathbf{z}_2$: aligning base direction with end-effector motion. In mobile manipulation tasks, especially in the case of differential drive bases, lateral movement cannot be achieved because of insufficient drive. Consequently, the task may fail if the robot reaches a configuration that requires sideways movement of the mobile base to continue. To avoid the risk, it is best to align the direction of the end-effector's motion with the base's forward or backward direction. This alignment can maximize the likelihood that the task will not fail because of insufficient drive when the base needs to move to reach a target position.

The expression (2.9) measures how far the base heading deviates from the direction of the end-effector movement. When $\theta = 0$, the base moves exactly in line with the end-effector.

Since the base velocity direction is controlled via its angular velocity ω_b, a correction strategy is introduced through the null-space velocity component:

$$\mathbf{z}_2 = -\alpha_2 \, \theta \cdot \mathbf{e}_\theta - \alpha_3 \int_0^t \theta(\tau) \, \mathrm{d}\tau \cdot \mathbf{e}_\theta \tag{4.36}$$

where $\alpha_2 > 0$ and $\alpha_3 > 0$ are tunable gains, and $\mathbf{e}_\theta \in \mathbb{R}^8$ is a selector vector defined as follows:

$$\mathbf{e}_\theta = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T \tag{4.37}$$

The vector $\mathbf{e}_\theta$ penalizes yaw deviation by assigning 1 to the base yaw joint q_2, thereby generating a corrective torque for reorientation. (4.36) is a PI controller with respect to θ. If only the instantaneous angle error θ is used, the reference alignment may lag. Introducing the integral term helps accumulate past orientation errors and gradually eliminate steady-state misalignment, achieving more consistent convergence even in the presence of delayed response.

Final control law

The composite control command becomes:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_{\text{task}} + \dot{\mathbf{q}}_{\text{null}} = \mathbf{J}_W^{*+}\mathbf{v}_{\text{ref}} + \left(\mathbf{I} - \mathbf{J}^{*+}\mathbf{J}^*\right)(\mathbf{z}_1 + \mathbf{z}_2) \tag{4.38}$$

where:

- $\dot{\mathbf{q}}_{\text{task}} = \mathbf{J}_W^{*+}\mathbf{v}_{\text{ref}}$: primary task velocity ensuring the desired end-effector velocity $\mathbf{v}_{\text{ref}} \in \mathbb{R}^6$,
- $\dot{\mathbf{q}}_{\text{null}} = (\mathbf{I} - \mathbf{J}^{*+}\mathbf{J}^*)(\mathbf{z}_1 + \mathbf{z}_2)$: secondary objectives projected in the null space of the Jacobian.

The null-space components are defined as follows:
Distance regulation:

$$\mathbf{z}_1 = -\alpha_1(d - d_{\text{target}})\mathbf{J}_d^T \tag{4.39}$$

where:

- $d = \sqrt{(x_e - x_b)^2 + (y_e - y_b)^2}$: horizontal distance between end-effector and base,
- $\mathbf{J}_d$: Jacobian of d with respect to generalized coordinates $\mathbf{q}$, given by:

$$\mathbf{J}_d = \frac{1}{d}\begin{bmatrix} dx & dy \end{bmatrix}\left(\begin{bmatrix} \mathbf{J}_x^* \\ \mathbf{J}_y^* \end{bmatrix} - \begin{bmatrix} 0 & \cdots & 0 & \cos\theta_r \\ 0 & \cdots & 0 & \sin\theta_r \end{bmatrix}\right) \tag{4.40}$$

with $dx = x_e - x_b$, $dy = y_e - y_b$, and θ_r denoting the base orientation.
Base-end-effector alignment:

$$\mathbf{z}_2 = -\alpha_2\,\theta \cdot \mathbf{e}_\theta - \alpha_3 \int_0^t \theta(\tau)\,\mathrm{d}\tau \cdot \mathbf{e}_\theta \tag{4.41}$$

where:

- θ: angle between base velocity and end-effector velocity,
- $\mathbf{e}_\theta = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$: unit vector selecting the rotational velocity of the base in $\mathbf{q}$.

5

Online Estimator

While the previous chapters presented a whole-body control scheme capable of executing coordinated motions, the controller assumes that certain kinematic properties of the mechanism, such as motion direction and curvature, are available. This chapter describes an online estimator that enables the system to autonomously learn these parameters during interaction. Rather than relying on predefined or manually tuned constraint information, the estimator progressively updates its internal representation of the mechanism's motion characteristics by observing the end-effector velocity.

The following update laws introduced in [3] are used to estimate the direction of motion $\hat{\mathbf{x}}_h$ and the curvature vector $\hat{\boldsymbol{\kappa}}$ of the mechanism:

$$\dot{\hat{\mathbf{x}}}_h = -\gamma v_{\text{ref}} \mathbf{P}(\hat{\mathbf{x}}_h)\mathbf{v}_f - v_{\text{ref}} \hat{\mathbf{x}}_h \times \hat{\boldsymbol{\kappa}} \tag{5.1}$$

where:

- $\gamma > 0$: adaptation gain determining convergence speed.
- $\hat{\mathbf{x}}_h \in \mathbb{R}^3$: unit vector estimating the motion direction of the handle.
- $v_{\text{ref}} \in \mathbb{R}$: reference velocity magnitude along the estimated motion direction, defined below.
- $\mathbf{P}(\hat{\mathbf{x}}_h) = \mathbf{I}_3 - \hat{\mathbf{x}}_h \hat{\mathbf{x}}_h^T$: projection matrix onto the orthogonal complement of $\hat{\mathbf{x}}_h$.
- $\mathbf{v}_f \in \mathbb{R}^3$: PI force feedback control term compensating the constraint directions.
- $\hat{\boldsymbol{\kappa}} \in \mathbb{R}^3$: estimate of the rotational axis scaled by curvature.

The adaptation law for the curvature vector is given by:

$$\dot{\hat{\boldsymbol{\kappa}}} = \Gamma_{\kappa}\, \hat{\mathbf{x}}_h \times \mathbf{v}_{\text{ref}}, \tag{5.2}$$

where:

- $\Gamma_{\kappa} \in \mathbb{R}^{3\times 3}$: a positive-definite matrix determining the curvature adaptation rate.
- $\mathbf{v}_{\text{ref}} \in \mathbb{R}^3$: reference velocity vector of the end-effector given by an adaptive velocity controller based on $\mathbf{v}_f$ and $\hat{\mathbf{x}}_h$.

To resolve the directional ambiguity, the signed magnitude of the velocity along $\hat{\mathbf{x}}_h$ is computed as follows:

$$v_{\text{ref}} = \text{sgn}\left(\hat{\mathbf{x}}_h^T \mathbf{v}_{\text{ref}}\right) \|\mathbf{v}_{\text{ref}}\|. \tag{5.3}$$

where:

- $\text{sgn}(\cdot)$: sign function indicating alignment of $\hat{\mathbf{x}}_h$ and $\mathbf{v}_{\text{ref}}$.
- $\|\mathbf{v}_{\text{ref}}\|$: Euclidean norm of the reference velocity vector.

The estimator updates two time-varying quantities: (i) the unit vector $\hat{\mathbf{x}}_h(t)$ that tracks the true motion direction of the handle (i.e., where the end-effector should go), and (ii) the vector $\hat{\boldsymbol{\kappa}}(t)$ indicates the axis of rotation and the curvature (the curvature vanishes in the prismatic motion).

Step 1 - determine the moving direction (Eq. (5.3)**).**

$$v_{\text{ref}} = \text{sgn}\big(\hat{\mathbf{x}}_h^T \mathbf{v}_{\text{ref}}\big)\ \|\mathbf{v}_{\text{ref}}\|$$

This scalar redefinition ensures that the positive direction of the command magnitude v_{ref} is always consistent with the current direction estimate $\hat{\mathbf{x}}_h$.

Step 2 - direction update (Eq. (5.1)**).**

$$\dot{\hat{\mathbf{x}}}_h = -\gamma v_{\text{ref}} \mathbf{P}(\hat{\mathbf{x}}_h)\mathbf{v}_f - v_{\text{ref}} \hat{\mathbf{x}}_h \times \hat{\boldsymbol{\kappa}}$$

- The projector $\mathbf{P}(\hat{\mathbf{x}}_h) = \mathbf{I} - \hat{\mathbf{x}}_h \hat{\mathbf{x}}_h^T$ removes the component of the force-feedback vector $\mathbf{v}_f$ that is parallel to $\hat{\mathbf{x}}_h$, leaving only the constraint directions and $-\gamma v_{\text{ref}}$ determines the speed of convergence. By reducing the force in the constraint direction, the end-effector and the mechanism are prevented from resisting each other, so that $\hat{\mathbf{x}}_h$ converges to the true direction.

- The cross product $\hat{\mathbf{x}}_h \times \hat{\boldsymbol{\kappa}}$ lies in the tangent space of the unit sphere $\mathbb{S}^2$. Therefore, adding (and scaling) this term will keep $\|\hat{\mathbf{x}}_h\| = 1$ time-invariant, thus preventing numerical drift in the estimated norm.

Step 3 - curvature update (Eq. (5.2)).

$$\dot{\hat{\boldsymbol{\kappa}}} = \Gamma_{\kappa} \left(\hat{\mathbf{x}}_h \times \mathbf{v}_{\text{ref}} \right).$$

Because the inner product $\hat{\mathbf{x}}_h^T \mathbf{v}_{\text{ref}} = v_{\text{ref}}$, the vector $\hat{\mathbf{x}}_h \times \mathbf{v}_{\text{ref}} = \hat{\mathbf{x}}_h \times (v_{\text{ref}} \hat{\mathbf{x}}_h) = \mathbf{0}$ when $\hat{\mathbf{x}}_h$ is aligned with the true direction. Otherwise, it produces a rotation of $\hat{\boldsymbol{\kappa}}$ around the instantaneous error axis so that, once convergence of $\hat{\mathbf{x}}_h$ is achieved, $\hat{\boldsymbol{\kappa}}$ also aligns with its physical counterpart $\boldsymbol{\kappa}$.

In summary, the estimator "listens" to the force/torque feedback: any velocity component that causes a non-zero force on the motion direction's subspace is interpreted as the end-effector interacting with the mechanism in the wrong direction, and is therefore used to rotate $\hat{\mathbf{x}}_h$ (and indirectly $\hat{\boldsymbol{\kappa}}$) until the unwanted force disappears. For a detailed mathematical proof, see the original paper [3].

Implementation in the Absence of Force Feedback

Due to time limitations, force/torque sensors are not used in this thesis. Instead, we adopt the control structure illustrated in Figure 2.3b, which bypasses the need for force feedback.

Instead of using the PI force feedback term $\mathbf{v}_f$ defined in (5.1) to generate the reference velocity $\mathbf{v}_{\text{ref}}$, we directly specified $\mathbf{v}_{\text{ref}}$ based on prior knowledge of the task constraints, such as the assumed revolving axis of the door. This predefined $\mathbf{v}_{\text{ref}}$ is aligned with the actual unconstrained motion direction of the mechanism, which ensures the end-effector will always move along the correct trajectory.

Formally, we replace (5.1) – (5.3) with:

$$\dot{\hat{\mathbf{x}}}_h = -\gamma v_{\text{ref}} \mathbf{P}(\hat{\mathbf{x}}_h) \mathbf{v}_{\text{ref}}^* - v_{\text{ref}} \hat{\mathbf{x}}_h \times \hat{\boldsymbol{\kappa}} \tag{5.4}$$

$$\dot{\hat{\boldsymbol{\kappa}}} = \Gamma_{\kappa} \hat{\mathbf{x}}_h \times \mathbf{v}_{\text{ref}}^* \tag{5.5}$$

$$v_{\text{ref}} = \text{sgn}\left(\hat{\mathbf{x}}_h^T \mathbf{v}_{\text{ref}}^* \right) \|\mathbf{v}_{\text{ref}}^*\| \tag{5.6}$$

where $\mathbf{v}_{\text{ref}}^*$ is the predefined velocity that meets the constraints.

This simplification allows us to isolate and test the estimation dynamics of the update laws. In particular, we compare the estimated motion direction $\hat{\mathbf{x}}_h$ and curvature vector $\hat{\boldsymbol{\kappa}}$ against the true constraint parameters we pre-assumed.

This setup enables us to evaluate the convergence properties of the estimators under ideal motion conditions and their ability to correctly identify the underlying constraint geometry of the mechanism.

6

Results and Discussion

This chapter presents simulation and real-world experimental results using a Heron robot that integrates a 6-DOF UR5e manipulator and a differentially driven MiR base, as is shown in Figure 6.1. The new joint velocities were calculated at 500 Hz, and commands are sent to the robot at 500 Hz in simulation and 50 Hz (approximate value) in actual experiments. Throughout the process, the robot first navigates to a predefined pose in front of the mechanism and then attempts to execute a predefined trajectory with the end-effector to simulate an opening motion.

Figure 6.1 Heron robot

6.1 Navigation to Pre-Interaction Pose

To evaluate the navigation performance, we analyze the tracking errors of position and orientation during the trajectory from the initial configuration $[0\,\text{m}, 0\,\text{m}, 0\,\text{rad}]$ to the goal position $[-1.5\,\text{m}, -1.15\,\text{m}, 0\,\text{rad}]$, as

shown in Figure 6.2. This goal corresponds to a displacement in the x- and y-directions while maintaining the same orientation, indicating that the robot is required to translate without rotating. The velocity gains are set as follows: $k_1 = k_3 = 1.0$, $k_2 = 2.0$.

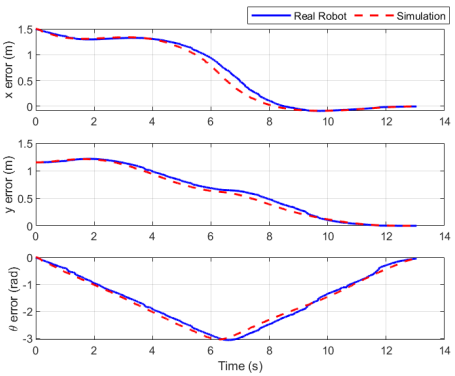


Figure 6.2 During navigation, the errors in x, y, and θ are plotted as three separate curves, with the blue lines representing the real robot and the red dashed lines representing the simulation results.

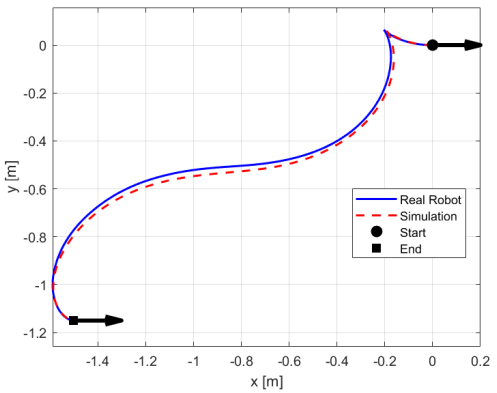


Figure 6.3 The trajectory of the base during navigation; the arrows at the start and end positions indicate the robot's starting and final orientations.

Figure 6.3 shows the trajectory of the base during navigation. We can divide the navigation process into the following stages:

- **0 – 2 s:** In the initial stage, the control law generates a small negative linear velocity v, which causes the robot to adjust the heading angle by reversing a short distance and point its nose toward the final position.
- **2 – 11 s:** As the robot follows the S-shaped trajectory, the errors in x and y begin to decrease rapidly. Initially, the robot turns right, following approximately a quarter circle. During this phase, the orientation error increases and peaks at about 180 degrees at about 6.5 seconds — indicating that the robot is temporarily facing the opposite direction of the target heading. The robot then turns left, completes another quarter circle, and eventually reaches its destination.
- **11 – 13 s:** When the robot reaches the target, ρ becomes small enough, leading the linear velocity term in the control law to diminish. Thereby, the robot effectively reduces the orientation error to zero through in-place rotation.

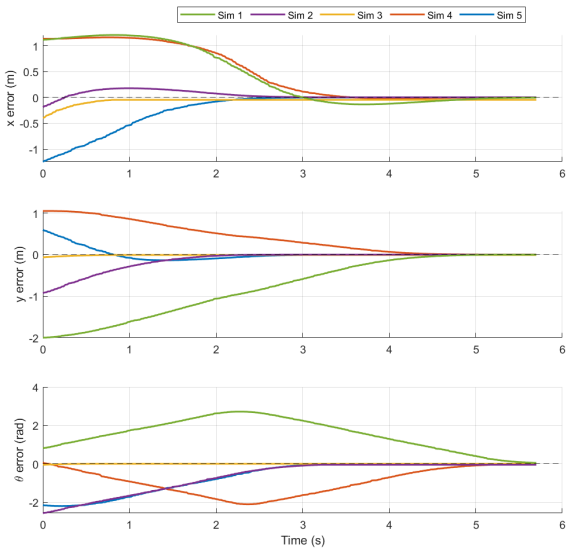


Figure 6.4 Errors in x, y, and θ for different target configurations.

Due to laboratory space limitations, we only performed more tests in simulation. Figure 6.4 verifies that the robot can always reach the target position under different target configurations. The results validate the controller design given by (3.5) and (3.6). The absence of oscillations in the direction tracking indicates that the nonlinear damping term $\frac{\sin\gamma\cos\gamma}{\gamma}$ suppresses overtuning when γ is large and linearizes the controller around $\gamma = 0$, thus stabilizing the angular motion. In summary, the results show that the proposed navigation controller is capable of reliably driving the robot to the desired pose with high accuracy.

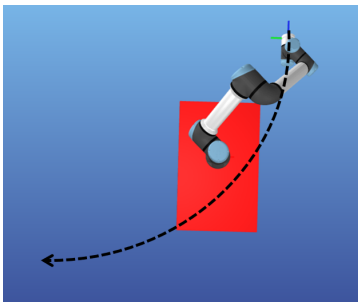
6.2 Whole Body Control

For both simulation and real-world experiments, we used the control framework in Figure 2.3b, i.e., without force feedback and the adaptive velocity controller. We focus on evaluating the performance of the proposed full-body controller in four representative 1-DOF mechanisms, as is demonstrated in Figure 6.5:

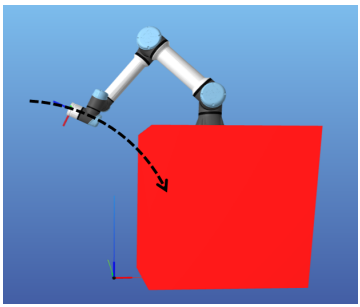
- Type I: Rotational door with vertical axis (common door).
- Type II: Rotational door with horizontal axis (e.g., oven door).
- Type III: Sliding door with lateral translation.
- Type IV: Pull-out drawer.

The end-effector is controlled to follow a predefined trajectory by assigning it a velocity in its local frame $\{E\}$. There are two main benefits to this strategy. First, trajectory generation is straightforward and reliable since circular motion is the natural result of a constant local velocity. Second, the end-effector rapidly deviates from the intended route if its pose deviates because the velocity is applied in the mismatched local frame. This error amplification makes it easier to assess the performance of the whole-body controller.

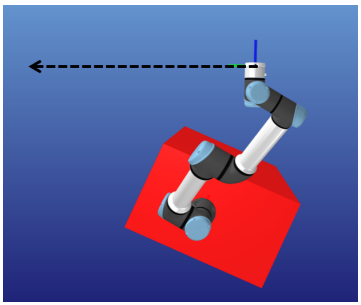
The simulation parameters are set as follows: $d_{target} = 0.75$ m, $\alpha_1 = 20$, $\alpha_2 = 2$, $\alpha_3 = 0.1$.



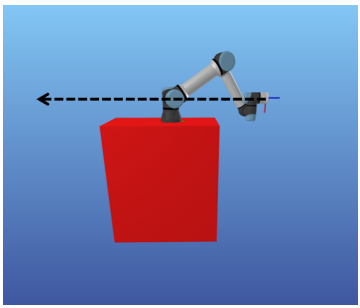
(a) Rotational door with vertical axis



(b) Rotational door with horizontal axis



(c) Sliding door



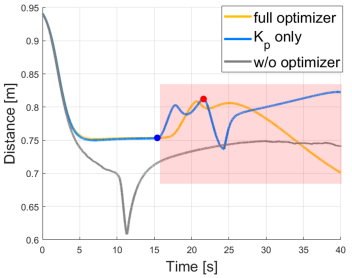
(d) Pull-out drawer

Figure 6.5 Four different types of 1-DOF mechanisms and the desired trajectories.

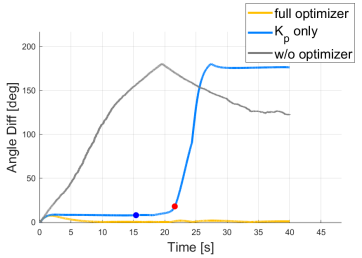
Type I: Rotational door with vertical axis

The rotation radius is set to 0.8 m.

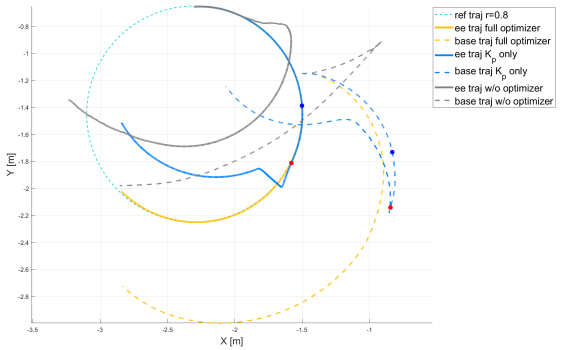
Figure 6.6 demonstrates the effectiveness of different null-space optimizers in regulating the coordinated motion between the base and the end-effector during door-opening tasks. When the full optimizer is employed, the end-effector follows the desired trajectory closely throughout the entire motion, confirming the correctness of the extended Jacobian computation and the effectiveness of the optimizer. In contrast, without the angular alignment optimizer $\mathbf{z}_2$, the non-holonomic constraint of the differential-drive base causes the manipulator to quickly reach joint limits. As a result, the end-effector deviates from the trajectory. Despite the presence of the distance-maintaining optimizer $\mathbf{z}_1$, the base-to-end-effector distance still drops below the safety threshold.



(a) Distance between the base and the end-effector projected onto the *xy* plane.



(b) Orientation difference between the base heading and the end-effector direction.

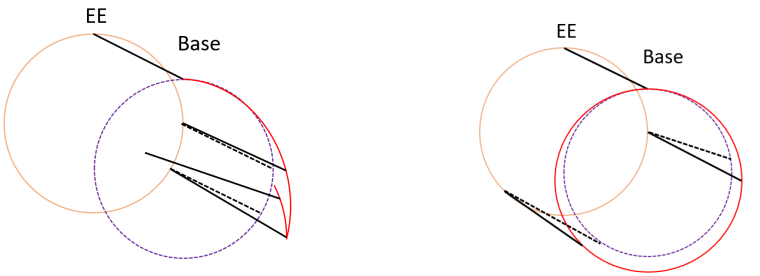


(c) Trajectories of the end-effector and the base in the *xy* plane.

Figure 6.6 Comparison of end-effector motion, base distance, and orientation alignment under three control strategies with $r = 0.8\,\text{m}$: (1) full optimizer, (2) with angle alignment optimizer but only proportional control, and (3) no angle alignment optimizer (distance optimizer included in all cases). Markers with the same indicate key positions at specific time points when using proportional control only.

Figure 6.7 compares the actual trajectories of the base and the end-effector under two configurations of the optimizer $\mathbf{z}_2$. The end-effector trajectory is shown in yellow, while the base trajectory is shown in red. A purple dashed curve represents the ideal trajectory that the base would follow if the direction and magnitude of its velocity are perfectly the same as those of the end-effector. At several instances, a black line connects the base and the end-effector positions: dashed lines show the ideal situation, when the distance is constant, while solid lines show the actual system be-

havior. The instantaneous distance between the base and the end-effector is reflected in the length of each line.



(a) **(b)**

Figure 6.7 Comparison of the actual trajectories of the base and the end-effector under two configurations of the optimizer $\mathbf{z}_2$: one using only a proportional term (left), and the other using the complete controller with both proportional and integral components (right).

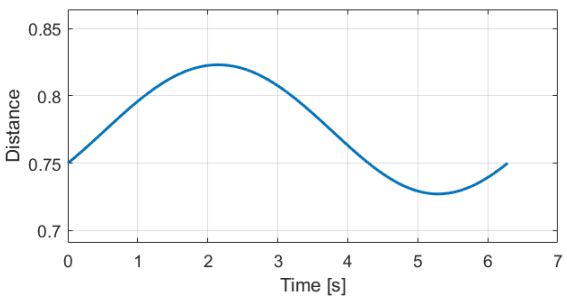


Figure 6.8 Distance between the base and the end-effector when the yaw correction $\mathbf{z}_2$ includes an integral term. A small velocity mismatch causes near-sinusoidal oscillations in the distance.

When $\mathbf{z}_2$ contains only proportional terms, the trajectory is tracked well initially, but then still drifts. Ideally, both the base and the end-effector should follow circular trajectories with the same radius and angular velocity. However, under purely proportional control, a steady-state angular error of about 8° remains between the two velocity directions (Figure 6.6b), leading to increasing trajectory divergence. Before the blue marker, $\mathbf{z}_1$ still maintains a safe distance by adjusting the speed of the base. However, after

reaching the blue marker, the end-effector is on the side of the base, leading to the base losing control of the distance because it cannot move sideways. Between the blue marker and red marker, the manipulator compensates for the growing distance by stretching, but after the red marker, joint limits are hit, resulting in tracking failure, as visualized in Figure 6.6c.

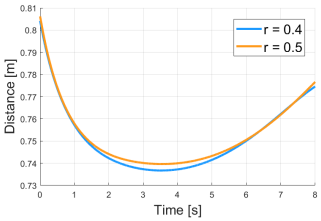
The residual angular error is eliminated by introducing an integral term in the yaw correction of $\mathbf{z}_2$, so that the base and end-effector velocities are fully aligned. At this time, the trajectory of the base is circular, and in order to maintain a constant distance, the trajectories of the base and the end-effector must have equal radii, which means the same linear velocity in the world frame $\{W\}$. However, the inverse kinematics based on the Jacobian matrix prioritizes the matching of angular velocities rather than the matching of spatial velocities. Figure 6.7b shows the trajectory change caused by a 6% velocity deviation. Figure 6.8 shows that despite the small deviation, the distance change still shows obvious near-sinusoidal oscillations, which can also be observed in the red area of Figure 6.6a. However, although the distance could not be preserved, the end-effector keeps moving on the set path, which confirms that the null-space optimization framework can maintain the integrity of the primary task even when some secondary goals cannot be achieved.

Type II: Rotational Door with Horizontal Axis

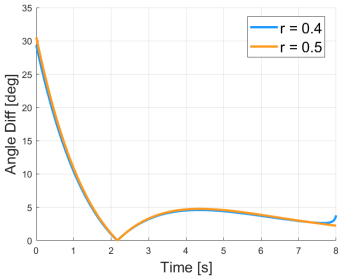
The rotation radius is set to 0.4 m and 0.5 m.

Figure 6.9 shows the motion at different radii. Due to the limited length of the Heron robot arm, the end-effector is easy to collide with the base when moving downward, so its range of motion is very limited. This is illustrated in Figure 6.10, where the points where collisions are most likely to occur are marked.

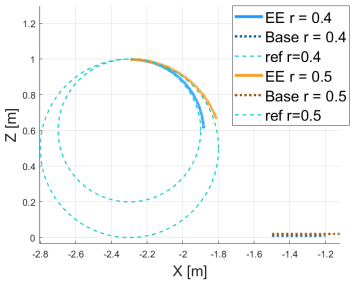
In Figure 6.9b, the sharp corner at about 2 corresponds to the overshoot introduced by the PI controller. Figure 6.9a shows that the distance between the base and the end-effector fails to converge to the expected value of 0.75 m. The reason for this phenomenon is similar to which in Type I, that there is a conflict between the secondary task and the primary task. In this case, the null-space optimization prioritizes the primary task and sacrifices the secondary goal when necessary. Nevertheless, the end-effector can successfully follow the desired trajectory under both radius settings.



(a) Distance between the base and the end-effector projected onto the *xy* plane.



(b) Orientation difference between the base heading and the end-effector direction.



(c) Trajectories of the end-effector and the base in the *xz* plane.

Figure 6.9 Comparison of end-effector motion, base distance, and orientation alignment using full optimizer with $r = 0.4\,\text{m}$ and $0.5\,\text{m}$.

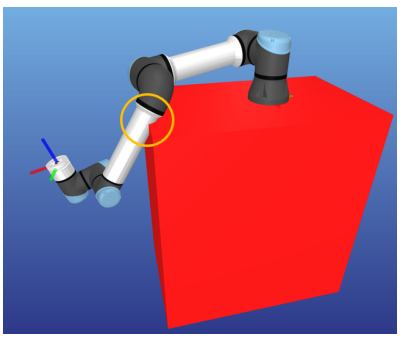


Figure 6.10 The yellow circle shows the most likely collision position.

Type III & IV: Sliding Door and Pull-out Drawer

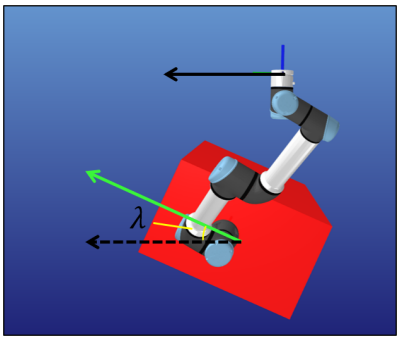


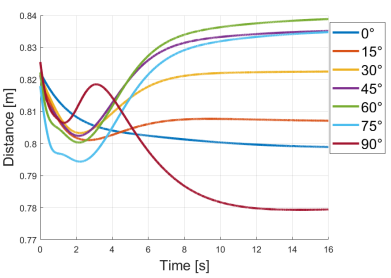
Figure 6.11 The angle λ between the base heading and the desired trajectory.

Since in both Type III and Type IV scenarios the end-effector moves along a straight line in the *xy* plane, they can be discussed together. The experimental results reveal a key factor that affects the performance: the initial angle λ between the base heading and the desired trajectory, as shown in Figure 6.11.

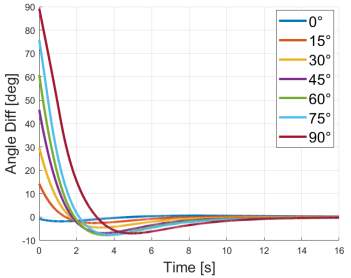
Figure 6.12 shows the motion of the end-effector and the base for different values of λ. As shown in Figure 6.12a, regardless of the initial angle λ, the distance between the base and the end-effector cannot converge to the desired value. This is primarily because of the non-holonomic constraints imposed by the differential-drive base, which limit the motion of the end-effector that remains almost constantly at the lateral side of the base. Nevertheless, once the motion stabilizes, the end-effector's velocity is predominantly contributed by the translational velocity of the base, while the manipulator joints remain stationary. As a result, there is no risk of collision after the system reaches steady-state motion.

From Figure 6.12c, it can be seen that when $\lambda < 45°$, the end-effector can accurately move along the desired path. However, as λ increases, the deviation between the actual trajectory and the desired trajectory increases significantly. It can be further observed that the trajectory of the base remains linear regardless of λ. This means that the base always rotates in place before moving forward to align its heading with the direction of motion of the end-effector. In addition, in the second half of the motion, the base and the end-effector always move along parallel paths. These two facts indicate that all trajectory deviations are introduced during the initial heading adjustment of the base, and the larger the required adjustment

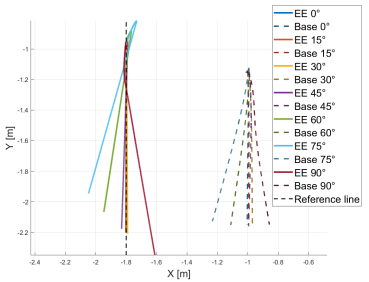
angle, the larger the error.



(a) Distance between the base and the end-effector projected onto the *xy* plane.



(b) Orientation difference between the base heading and the end-effector direction.



(c) Trajectories of the end-effector and the base in the *xy* plane.

Figure 6.12 Comparison of end-effector motion, base distance, and orientation alignment under varying initial heading angles. All runs use the full controller.

The root cause of this phenomenon lies in the misalignment between the rotation axes of the moblie base and the manipulator base, as is shown in Figure 6.13. Specifically, the projection of the vector ${}^{MB}\mathbf{t}_{AB}$ on the *xy* plane is not zero. Therefore, even a pure rotational motion of the base will introduce a linear velocity component of the end-effector in the *xy* plane. The advantage of this configuration is that the angular velocity ω of the base can contribute to the linear velocity planning of the end-effector, thereby improving reachability. However, when λ is large, the inverse kinematics solution requires too much compensation in the joint space to offset the linear velocity brought by ω, which may become infeasible. As a result,

the end-effector cannot move smoothly in the desired direction.

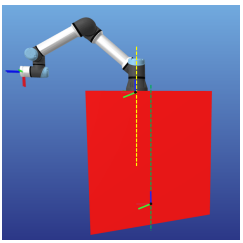
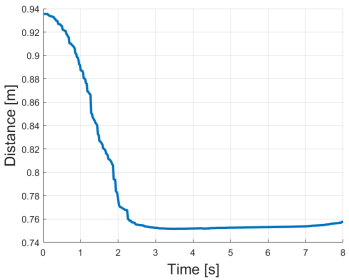
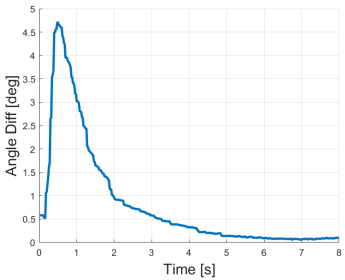


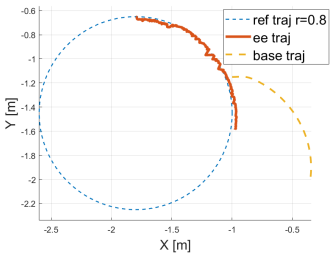
Figure 6.13 The rotation axes of the moblie base (green) and the manipulator base (yellow).



(a) Distance between the base and the end-effector projected onto the *xy* plane.

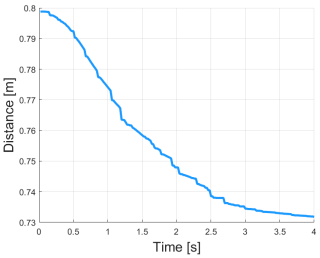


(b) Orientation difference between the base heading and the end-effector direction.

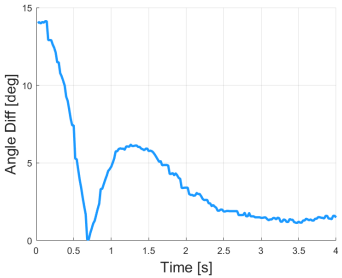


(c) Paths of the end-effector and the base in the *xy* plane.

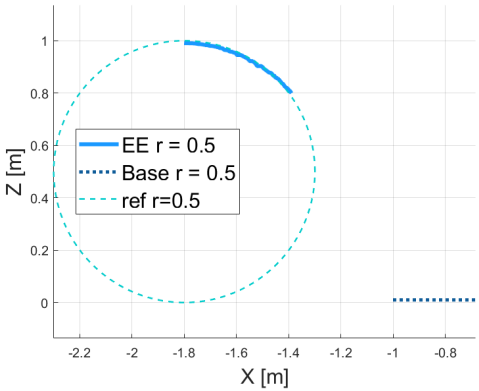
Figure 6.14 End-effector motion, base distance, and orientation alignment with $r = 0.8\,\text{m}$.



(a) Distance between the base and the end-effector projected onto the *xy* plane.



(b) Orientation difference between the base heading and the end-effector direction.



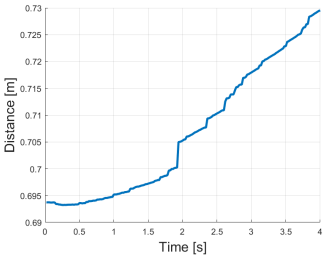
(c) Paths of the end-effector and the base in the *xz* plane.

Figure 6.15 End-effector motion, base distance, and orientation alignment with $r = 0.5\,\text{m}$.

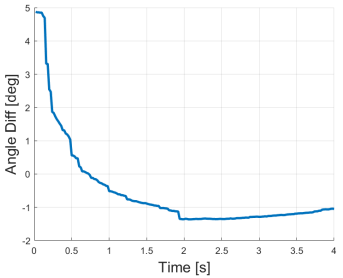
Real Robot Experiment

The parameters are set as follows: $d_{target} = 0.75$ m, $\alpha_1 = 20$, $\alpha_2 = 2$, $\alpha_3 = 1$.

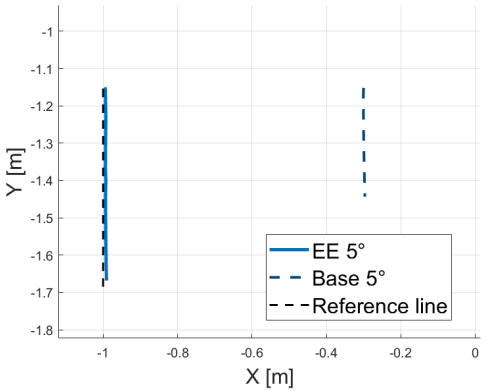
Figure 6.14 – Figure 6.16 show the experimental data of the actual robot running Type I – III trajectories, respectively. Compared with the simulation experiment, the operation of the robot is greatly limited by the control frequency.



(a) Distance between the base and the end-effector projected onto the *xy* plane.



(b) Orientation difference between the base heading and the end-effector direction.



(c) Paths of the end-effector and the base in the *xy* plane.

Figure 6.16 End-effector motion, base distance, and orientation alignment with initial heading angles $= 5°$.

The initial peak in the angle difference in Figure 6.14b is mainly because of the lower control frequency of the real robot (only one-tenth of the simulation frequency). The reduced frequency causes a significant delay in the base heading adjustment. As a result, the angle deviation increases temporarily in the initial stage. However, because of the integral component of the heading alignment controller, the system gradually accumulates the error and strengthens the correction torque over time. Eventually, the angle difference decays smoothly and finally converges.

When the initial heading error is large, this delay can not only result in

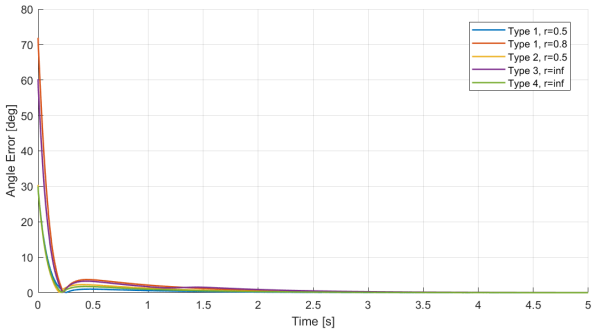
a slow response but can also cause the system to overshoot, causing the error to exceed the stabilization capacity of the proportional term itself. For example, the steady-state error observed in Figure 6.15b and the overshoot observed in Figure 6.16b illustrate this behavior. This performance degradation requires tighter control of the initial alignment between the robot's heading and the initial motion direction of the manipulator. Experimental results show that the robot will experience significant trajectory deviations in Type II and Type III tasks when the initial alignment error exceeds 15° and 10°, respectively.

Summary

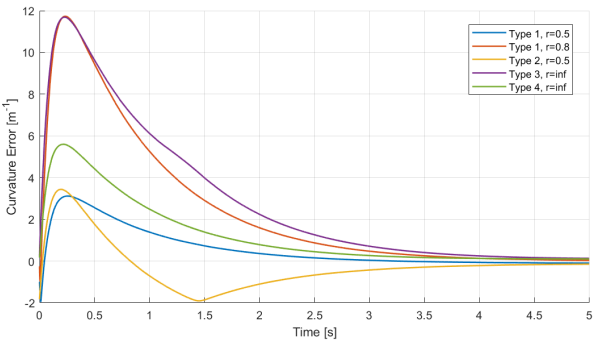
This section systematically evaluates the performance of the proposed whole-body control framework in four representative single-degree-of-freedom mechanisms. By assigning velocity commands in the local coordinate system of the end-effector, the method allows both intuitive trajectory generation and natural error amplification for performance evaluation. Extensive simulation results show that the null-space optimization strategy is effective in ensuring the execution of the primary task even in the presence of secondary objective conflicts (such as the distance adjustment from the base to the end-effector). This also reflects that the extended Jacobian matrix is calculated correctly. The results also highlight the limitations brought by non-holonomic constraints, especially when the base needs to make large heading corrections. Overall, the framework shows excellent adaptability and robustness in various operating scenarios.

6.3 Online Estimator

To evaluate the performance of the proposed online estimator, we monitor whether the estimated curvature $\hat{\kappa}$ and the estimated motion direction $\hat{\mathbf{x}}_h$ converge to the actual curvature κ and the desired direction $\mathbf{v}_{\text{ref}}$, respectively. The angle between the unit vectors $\hat{\mathbf{x}}_h$ and $\mathbf{v}_{\text{ref}}$ can be used as a metric to assess the alignment, while the curvature error is defined as $|\hat{\kappa}| - |\kappa|$. For Type I and Type II mechanisms (revolving doors), the true curvature is represented by the inverse of the radius, $|\kappa| = \frac{1}{r}$. For Type-III and Type IV mechanisms (sliding doors and drawers), the curvature is zero, i.e., $|\kappa| = 0$, corresponding to an infinite radius.



(a) Angle difference between estimated heading and reference velocity.



(b) Curvature error relative to desired radius.

Figure 6.17 Comparison of heading alignment and curvature convergence across motion types.

The plots in Figure 6.17 show the convergence behavior of the online estimator for the four different 1-DOF mechanism types. Figure 6.17a shows that the error between the estimated heading and the reference velocity direction always decreases to zero in all cases, and the convergence process is fast and smooth. Figure 6.17b shows that the curvature estimate $\hat{\kappa}$ decays correctly to zero, confirming that the estimator can reliably distinguish between curved and straight trajectories. Overall, the results verify that the proposed adaptive estimation scheme can ensure orientation alignment and curvature recognition in a variety of door and drawer mechanisms.

It is worth noting that even though the initial estimate of $\hat{\kappa}$ matches the

true value exactly, it still exhibits significant fluctuations before finally converging. This is mainly because of the large initial heading misalignment, which causes the deviation of $\hat{\mathbf{x}}_h$ to introduce curvature estimation errors through the cross-product term $\hat{\mathbf{x}}_h \times \mathbf{v}_{\text{ref}}^*$ in (5.5). Combining the two figures in Figure 6.17, it can be clearly seen that after about 0.25 seconds, $\hat{\mathbf{x}}_h$ is aligned with the true heading direction, thus eliminating the disturbing term and allowing $\hat{\kappa}$ to converge steadily to its true value. This interaction highlights the mutual influence between heading estimation and curvature correction, further validating the robustness of the proposed estimator in dealing with transient misalignment.

To sum up, the simulation results in Figure 6.17 demonstrate the effectiveness and robustness of the proposed online estimation framework. Specifically, Figure 6.17a confirms that the heading estimate $\hat{\mathbf{x}}_h$ consistently aligns with the reference direction $\mathbf{v}_{\text{ref}}$ in all cases, with a rapid and monotonic decay of angular error. Meanwhile, Figure 6.17b indicates that the curvature estimate $\hat{\kappa}$ successfully converges to the correct value, distinguishing between revolute and prismatic mechanisms.

7

Conclusion

By combining navigation, whole-body control, and online estimation, we extended previous work on fixed-base manipulators and created a unified system that allows differential-drive mobile manipulators to interact with one-degree-of-freedom (1-DoF) mechanisms.

We first developed a nonlinear feedback controller that guides the robot to an appropriate pre-interaction pose. Its ability to achieve precise convergence in both position and orientation, even in the presence of nonholonomic restrictions, was validated by simulation and hardware experiments.

We then created a whole-body control scheme for an integrated 6-DOF manipulator and a differentially-driven base using an extended Jacobian formulation. Through null-space optimization, the controller ensures the execution of the primary end-effector task while regulating secondary objectives, such as maintaining a safe base-arm distance and aligning the base heading with the end-effector motion. Simulation results based on four 1-DOF mechanisms show that the controller achieves robust and smooth coordination. In scenarios with conflicting secondary objectives, the framework prioritizes task feasibility while ensuring safety, thus verifying the correctness of the extended Jacobian and the effectiveness of the null-space design.

Finally, we integrated an online estimator to autonomously learn the mechanism's curvature and motion direction during interaction. The estimator reliably converged to the correct constraint parameters based on observed motion, validating its applicability in real-world deployments where sensing may be limited.

Overall, the proposed framework successfully addresses the key issues of mobile manipulation interacting with 1-DoF mechanisms. While the proposed framework demonstrates robust performance in simulation and

experiments, several promising directions remain open. First, integrating real force/torque feedback into the online estimator would allow the system to adapt more effectively under physical uncertainties and external disturbances. Second, extending the control scheme to multi-DoF mechanisms or to tasks involving sequential interactions (e.g., opening a door and manipulating objects behind it) would broaden applicability. Finally, incorporating learning-based components, such as reinforcement learning for secondary task optimization, could enhance adaptability to complex and cluttered environments.

Bibliography

[1] C. C. Kessens, J. B. Rice, D. C. Smith, S. J. Biggs, and R. Garcia. "Utilizing compliance to manipulate doors with unmodeled constraints". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 483–489.

[2] C. Ott, B. Bäuml, C. Borst, and G. Hirzinger. "Employing cartesian impedance control for the opening of a door: a case study in mobile manipulation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Mobile Manipulators*. 2005.

[3] Y. Karayiannidis, C. Smith, F. E. Viña, P. Ögren, and D. Kragic. "An adaptive control approach for opening doors and drawers under uncertainties". *IEEE Transactions on Robotics* **32**:1 (2016), pp. 161–175.

[4] G. Niemeyer and J.-J. Slotine. "A simple strategy for opening an unknown door". In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 1997, pp. 1448–1453.

[5] J. Haviland and P. Corke. "Neo: a novel expeditious optimisation algorithm for reactive motion control of manipulators". *IEEE Robotics and Automation Letters* **6**:2 (2021), pp. 1043–1050.

[6] J. Haviland and P. Corke. *A purely-reactive manipulability-maximising motion controller*. 2020. arXiv: 2002.11901 [cs.RO].

[7] G. Kang, H. Seong, D. Lee, and D. H. Shim. "A versatile door opening system with mobile manipulator through adaptive position-force control and reinforcement learning". *Robotics and Autonomous Systems* **180** (2024), p. 104760.

[8] B. Bayle, J.-Y. Fourquet, and M. Renaud. “Manipulability of wheeled mobile manipulators: application to motion generation”. *International Journal of Robotics Research* **22** (2003).

[9] B. Burgess-Limerick, C. Lehnert, J. Leitner, and P. Corke. “An architecture for reactive mobile manipulation on-the-move”. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 1623–1629.

[10] B. Burgess-Limerick, C. Lehnert, J. Leitner, and P. Corke. *Enabling failure recovery for on-the-move mobile manipulation*. 2023. arXiv: 2305.08351 [cs.RO].

[11] M. Suomalainen, Y. Karayiannidis, and V. Kyrki. “A survey of robot manipulation in contact”. *Robotics and Autonomous Systems* **156** (2022), p. 104224.

[12] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer, 2010.

[13] H. Arisumi, J.-R. Chardonnet, and K. Yokoi. “Whole-body motion of a humanoid robot for passing through a door: opening a door by impulsive force”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009, pp. 428–434.

[14] K. M. Lynch and F. C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, Cambridge, UK, 2017. ISBN: 9781107156302.

[15] O. Khatib. “A unified approach for motion and force control of robot manipulators: the operational space formulation”. *IEEE Journal on Robotics and Automation* **3**:1 (1987), pp. 43–53.

[16] C. W. Wampler. “Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods”. *IEEE Transactions on Systems, Man, and Cybernetics* **16**:1 (1986), pp. 93–101.

[17] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal. “Comparative experiments on task space control with redundancy resolution”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2005, pp. 3901–3908.

[18] A. De Luca, G. Oriolo, and P. Giordano. “Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2006, pp. 1867–1873.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
September 2025

Document Number
TFRT-6299

Author(s)
Yuyao Liu

Supervisor
Yiannis Karayiannidis, Dept. of Automatic Control, Lund University, Sweden
Björn Olofsson, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
Mobile Robot Manipulator Control for Interacting with 1-DoF Mechanisms

Abstract

This study investigates the interaction problem between a differentialdrive mobile manipulator and one-degree-of-freedom (1-DoF) mechanisms. While previous research has focused on stationary robotic platforms, this work puts emphasis on mobile interaction.

Building upon a previously proposed adaptive control strategy for fixed-base manipulators, a holistic control framework is developed to coordinate the motion of the mobile base and the manipulator. The adaptive control strategy is extended to mobile robots, enabling repositioning during interaction to achieve larger door opening angles and avoid collisions. A nonlinear velocity controller is designed to guide the differential-drive mobile base to a suitable position for initiating interaction. To achieve simultaneous motion between the mobile base and the manipulator, an extended Jacobian matrix formulation is employed, integrating both forward and inverse kinematics of the base and arm.

To maintain a safe distance between the manipulator and the mobile base, a null-space optimization method is integrated. By modifying the base orientation in accordance with the direction of motion, this method also compensates for the non-holonomic restrictions of differential-drive platforms. Additionally, to improve manipulability and prevent kinematic singularities during the door-opening job, an adaptive damping factor is incorporated. Furthermore, building upon previously recorded motion trajectories, an online estimator is introduced, allowing the robot to learn the rotational axis, the door's rotation radius, and the expected motion direction.

Simulation and real robot experiments are conducted to evaluate the suggested control framework in a range of door-opening scenarios. The results demonstrate that the proposed framework achieves robust coordination between the mobile base and manipulator, enabling safe and efficient interaction with various 1-DoF mechanisms. The findings confirm the applicability of the approach for real-world mobile manipulation tasks under kinematic uncertainties.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-59

Recipient's notes

Security classification

http://www.control.lth.se/publications/