

UCLA

UCLA Electronic Theses and Dissertations

Title

Control and Motion Planning for a Low-Inertia Multi-DoF Robotic Manipulator with Proprioceptive Actuators for Dynamic Manipulation

Permalink

<https://escholarship.org/uc/item/10w4q5ft>

Author

Noh, Donghun

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Control and Motion Planning for a Low-Inertia Multi-DoF Robotic Manipulator with
Proprioceptive Actuators for Dynamic Manipulation

A dissertation submitted in partial satisfaction
of the requirements for the degree Doctor of Philosophy
in Mechanical Engineering

by

Donghun Noh

2024

© Copyright by
Donghun Noh
2024

ABSTRACT OF THE DISSERTATION

Control and Motion Planning for a Low-Inertia Multi-DoF Robotic Manipulator with
Proprioceptive Actuators for Dynamic Manipulation

by

Donghun Noh

in Mechanical Engineering

University of California, Los Angeles, 2024

Professor Dennis Hong, Chair

Robotic manipulators are increasingly required to operate in dynamic, human-centric environments, undertaking tasks that range from delicate grasping to high-speed, high-impact manipulation. While conventional high-gear-ratio manipulators and position control strategies perform admirably in structured industrial contexts, they often prove inadequate for these more complex, unstructured domains. This dissertation addresses these shortcomings by integrating novel hardware, control, motion planning, and machine learning methodologies to realize more adaptable and robust robotic manipulation.

Central to this work is the deployment of a proprioceptive manipulator, whose inherent backdrivability, high speed, and precise torque control distinguish them from conventional manipulators. Their low-inertia, lightweight design facilitates safer and more responsive robot-environment interactions. To fully exploit these advantages, we develop control methodologies that emphasize torque control rather than position control. In parallel, we introduce motion planning strategies that minimize jerk and incorporate dynamic constraints, thereby producing trajectories that are both efficient and smooth.

This dissertation further enhances adaptability through machine-learning approaches tailored for friction-aware grasping and dynamic manipulation. By constructing a friction coefficient dataset and training regression models, the system can efficiently adjust grasping forces in real-time. Additionally, ongoing research in imitation learning, drawing on human demonstrations and incorporating both visual data and physical models, shows promise in enabling the robot to eventually replicate human-level skill in complex, dynamically varying tasks.

Comprehensive evaluations include successful demonstrations of autonomous cooking and friction-aware grasping, as well as ongoing research into dynamic box-receiving tasks. While further validation and optimization remain active areas of study, results so far suggest that, when integrated with torque control, optimized motion planning, and model-based data-driven learning, the proposed proprioceptive manipulator can significantly enhance manipulation speed, stability, energy efficiency, and compliance. These findings provide a strong foundation for more intuitive, versatile, and collaborative robotic systems, better suited to the demands of real-world, human-centered environments.

The dissertation of Donghun Noh is approved.

Jonathan Hopkins

M. Khalid Jawed

Veronica Santos

Dennis Hong, Committee Chair

University of California, Los Angeles

2024

To my family.

TABLE OF CONTENTS

1	Introduction	1
1.1	Motivations and Objectives	1
1.2	Background	5
1.2.1	Manipulator Design	5
1.2.2	Manipulator Control	6
1.2.3	Manipulator Motion Planning	9
1.2.4	Machine Learning for Manipulation	13
1.3	Thesis Organization	17
2	Development, Motion Planning, and Control of a Proprioceptive Manip- ulator	19
2.1	Hardware Design	19
2.1.1	Existing Platforms	19
2.1.2	Design Overview	21
2.1.3	Prototype: Dual-Arm Manipulator With High Gear Ratio Servo Motor	22
2.1.4	Dual-Arm Manipulator Equipped Proprioceptive Actuators	25
2.2	Modeling	30
2.2.1	Definition	30
2.2.2	Kinematics	32
2.2.3	Dynamics	39
2.3	Control Strategies	39
2.3.1	Inverse Dynamics Controller Implementation	40

2.3.2	Impedance Controller Implementation	50
2.3.3	Jerk-Optimized Dynamic Model Predictive Control	51
2.4	Motion Planning: Time-Optimal and Jerk-Minimized Trajectory Planning	55
2.4.1	Trajectory Optimization Formulation	55
2.4.2	Experimental Results	58
2.5	Summary	69
3	YORI: Autonomous Cooking System Utilizing a Modular Robotic Kitchen and a Dual-Arm Proprioceptive Manipulator	70
3.1	System Design	70
3.2	Implementation: Manipulator Control and Motion Planning	76
3.3	Summary	77
4	Friction-Aware Grasping	80
4.1	Dataset Preparation and Processing	81
4.1.1	Friction Measurement Device	81
4.1.2	Data Collection Process	82
4.1.3	Validation of Dataset Utility	85
4.2	Model Architecture and Training	88
4.3	Training Results	90
4.4	Friction-Aware Grasping	93
4.4.1	Friction-Aware Grasping Algorithm	93
4.4.2	Experimental Validation	95
4.5	Summary	97

5	Adaptive Video Imitation for Dynamic Manipulation	98
5.1	Overview	98
5.1.1	Box Receiving as a Representative Dynamic Manipulation Task . . .	99
5.1.2	Our Approach: Translating Human Demonstrations into Robot Dy- namics	100
5.1.3	Control Strategies	101
5.2	Dataset Preparation	101
5.2.1	Data Collection	101
5.2.2	Data Processing	103
5.3	Visuomotor Policy Learning	109
5.3.1	Problem Setup for Visuomotor Policy Learning	109
5.3.2	Diffusion Policy	109
5.3.3	Latent Plan Transformer	110
5.4	Experiments and Results	111
5.5	Summary	113
6	Conclusion and Future Works	115
6.1	Conclusion	115
6.2	Future Plans	116
	References	121

LIST OF FIGURES

2.1	(a) LIMS (b) Blue.	20
2.2	Configuration of the dual-arm.	21
2.3	Pose with fully extended arm in horizontal direction.	22
2.4	Prototype of the dual-arm manipulator.	24
2.5	4-Bar linkage.	25
2.6	Forearm.	25
2.7	Upperarm.	26
2.8	Entire arm.	26
2.9	BEAR series.	28
2.10	Dual-arm manipulation platform equipped proprioceptive actuators.	29
2.11	Coordinate axis diagram: left view.	31
2.12	Coordinate axis diagram: front view.	31
2.13	Left arm frame assignment for DH parameters.	32
2.14	Repeatability test results based on motion capture tracking.	41
2.15	Tracking performance experiment sequence.	43
2.16	Result: end-effector x-axis / 1 rad/s / 0kg.	43
2.17	Result: end-effector y axis / 1 rad/s / 0kg.	44
2.18	Result: end-effector z-axis / 1 rad/s / 0kg.	44
2.19	Result: end-effector x-axis / 1 rad/s / 1kg.	44
2.20	Result: end-effector y axis / 1 rad/s / 1kg.	45
2.21	Result: end-effector z-axis / 1 rad/s / 1kg.	45
2.22	Result: end-effector x-axis / 1 rad/s / 2kg.	45

2.23	Result: end-effector y axis / 1 rad/s / 2kg.	46
2.24	Result: end-effector z-axis / 1 rad/s / 2kg.	46
2.25	Result: end-effector x-axis / 3 rad/s / 0kg.	46
2.26	Result: end-effector y axis / 3 rad/s / 0kg.	47
2.27	Result: end-effector z-axis / 3 rad/s / 0kg.	47
2.28	Result: end-effector x-axis / 3 rad/s / 1kg.	47
2.29	Result: end-effector y axis / 3 rad/s / 1kg.	48
2.30	Result: end-effector z-axis / 3 rad/s / 1kg.	48
2.31	Result: end-effector x-axis / 3 rad/s / 2kg.	48
2.32	Result: end-effector y axis / 3 rad/s / 2kg.	49
2.33	Result: end-effector z-axis / 3 rad/s / 2kg.	49
2.34	Comparison of joint positions between the reference trajectory and the MPC-generated trajectory.	53
2.35	Comparison of joint torques between the reference trajectory and the MPC-generated trajectory.	54
2.36	SE(3) quintic spline path and corresponding joint path derived via inverse kinematics.	59
2.37	Time-parameterization of the reference joint path using TOPP-RA, respecting velocity and acceleration limits.	60
2.38	Time-optimal and jerk-minimized trajectory derived from the reference path, balancing execution time and smoothness.	61
2.39	Minimum jerk trajectory obtained by applying the minimum jerk formulation twice, enforcing zero velocity at the via point.	63

2.40	Optimized minimum jerk trajectory that passes through the via point without enforcing zero velocity, resulting in a continuously smooth profile.	64
2.41	Final torque profiles obtained from the various tested trajectories, illustrating their dynamic feasibility.	65
2.42	Final jerk profiles for the tested trajectories, demonstrating differences in motion smoothness.	66
2.43	Bar chart summarizing the final torque values across all trajectories, enabling quantitative comparison.	67
2.44	Bar chart summarizing the final jerk values for all trajectories, highlighting the benefits of our optimization strategy in minimizing jerk.	68
3.1	Overview of the YORI System: A centrally located dual-arm manipulator with proprioceptive actuators is surrounded by four modular kitchen units equipped with either customized or newly developed cooking tools and appliances. Shown below, from left to right, are dishes prepared by the system, including Steak Frites, Tomato Penne Pasta, Spicy Fried Chicken, and Brownie.	71
3.2	Layout of the cooking cell. The annotated components include: (A) storage shelf, (B) rotating pot, (C) dish trolley, (D) convection oven, (E) salamander broiler, (F) deep fryer, (G) pasta cooker, (H) spice dispenser, and (I) induction cooktop.	72
3.3	Overview of the integrated appliances and their corresponding tools forming the YORI automated kitchen cell. Depicted components include: (a) the spice dispenser with its ingredient dispensing modules, (b) the food processor and dicing chamber, (c) the rotating mixer with a detachable pot, (d) the salamander broiler, (e) the convection oven, (f) the induction cooktop, (g) the deep fryer and water boiler with their associated cooking basket, (h) the induction pan and its matching squeegee, and (i) a custom-designed induction pan paired with its specialized squeegee sweeping tool.	74

3.4	Flow diagram illustrating the control structure and messages used to control the integrated appliances in the kitchen cell. The High Level Scheduler communicates commands and statuses with the appliance manager through the systems shared memory. The appliance manager in turn communicates with each local microcontroller one at a time over the shared usb bus by sending commands tied to one of the unique microcontroller IDs.	75
3.5	Overview of the YORI System’s software architecture, highlighting interfaces critical for high-frequency operations like safety checks, state checks, and motor control, communicating at 1000Hz. Conversely, components not requiring high-frequency interactions, such as camera systems, task scheduler, and appliance control, function at 30Hz or 60Hz. To guarantee swift communication and enhance system stability, data processing is executed via shared memory. Remote commands for the system are accepted through WiFi, leveraging the ROS2 middleware, solely upon specific order reception.	76
3.6	Example of impedance controller usage. (1) Current end-effector height and (2) Desired goal position height.	77
3.7	Example of transporting and grasping motions during the preparation of Steak Frites.	78
3.8	Demonstration of the autonomous robotic cooking system at a convention in Seoul, Korea, showcasing its ability to continuously cook Steak Frites.	79
4.2	Measured friction coefficients for the materials in the SMDRA, including their mean values and standard deviations.	82
4.3	Example materials included in the dataset. Top row (from left):carpet, grass, wood, concrete; Bottom row (from left):sand, rubber, stainless steel, cardboard.	83

4.4	Examples of images captured by an active stereo camera and annotated using the COCO Annotator tool. The top row shows an RGB image (left) and a depth image (right) of a sample containing carpet, grass, concrete, rubber, plastic, and tile. The bottom row presents another RGB (left) and annotated depth image (right) of a sample composed of carpet, grass, stainless steel, wood, plastic, and tile.	84
4.5	Segmentation results for four example test cases. The materials involved include grass, concrete, rubber, plastic, tile, stainless steel, cardboard, carpet, and sand. Each column compares the raw image, ground truth, and segmentation outputs from FCN and U-Net using both RGB and RGB-D inputs.	87
4.6	UNet architecture [RFB15].	89
4.7	TransUNet architecture combining Transformer encoding and convolutional decoding [CLY21].	90
4.8	Representative U-Net output demonstrating strong boundary delineation and effective local feature recognition.	92
4.9	Representative TransUNet output illustrating improved friction coefficient estimation, likely due to enhanced global context modeling.	92
4.10	Illustration of the friction-aware grasping experiment setup. In (a), the gripper is shown before rotation, and (b) illustrates a rapid rotation scenario. This dynamic testing environment evaluates the gripper’s ability to adapt its grasping forces in real-time to changes in object orientation and the estimated friction coefficient.	95
4.11	Comparison of torque profiles under traditional position-based control and friction-aware force-based control for both 100g and 200g boxes. Even with a safety factor of 3 applied, the friction-aware approach significantly reduces both peak and overall torque, demonstrating reliable grasping performance across varying loads.	96

5.1	Pose estimation, and box segmentation results are overlaid on an RGB image. The detected keypoints demonstrate accurate alignment with the subject’s body structure.	105
5.2	Keypoint trajectories for the right-side joints (hip, shoulder, elbow, wrist, and finger). Each joint includes raw and filtered trajectories for comparison.	106
5.3	Simplified skeleton visualization focusing on key joints relevant to dynamic motion analysis.	107
5.4	Joint Information: plots of joint angles, velocities, accelerations, and torque for the selected joints over time. These dynamics are derived from filtered keypoint trajectories.	108
5.5	Overview of the Latent Plan Transformer (LPT) architecture. The latent plan z is sampled from the prior $p_\alpha(z)$ and conditions the Transformer-based policy generator p_β , which outputs a sequence of actions based on past actions, observations, and the latent plan. This figure illustrates how LPT integrates the latent plan sampling and policy generation processes within a single framework.	110
5.6	Simulation setup illustrating the environment in which the preliminary experiments were conducted.	112
6.1	Impact and force-based tasks, such as hammering, represent challenging real-world scenarios for mobile manipulators.	119
6.2	Painting tasks, requiring careful force modulation and dexterity, illustrate the potential for mobile manipulators in complex human-centric environments.	120

LIST OF TABLES

1.1	Specifications of Different Robotic Actuators	3
1.2	Summary of Trajectory Generation Methods	13
2.1	Torque Requirement for Each Pitch Actuator	23
2.2	Definition of the Coordinate Axes	30
2.3	Constant Parameters for Frames	33
2.4	DH Parameters of the Left Arm	34
2.5	Deviation Measurements Recorded at Designated Points	42
2.6	Deviation Measurements for Different Manipulators	42
4.1	Training Parameters for Semantic Segmentation Models	86
4.2	Performance Metrics for Semantic Segmentation Models	86
4.3	Mean MAE Results for U-Net and TransUNet	91
5.1	Preliminary Timing Results for Trajectory Generation and Replanning	113

ACKNOWLEDGMENTS

A few years ago, I could never have imagined pursuing a PhD, yet here I am, writing my dissertation as I complete this journey. My journey began in my senior year of university, which was a relatively late start to discovering my passion for building robots. Nonetheless, I fully committed myself to this field, with that passion as my driving force. Reflecting on this journey, I find the emotions and insights I've gained to be deeply meaningful and life-changing.

Dr. Dennis Hong, you have been an exceptional PhD advisor and an invaluable mentor. Your unwavering support has been instrumental in helping me navigate and complete this journey. Beyond your intellectual guidance, you created countless opportunities for me and fostered an environment of trust and encouragement that gave me the confidence to grow. You have also set a remarkable example of how to approach life and relationships with wisdom and professionalism. I have the utmost respect for you as both a pioneering roboticist and an extraordinary individual. I am deeply grateful for your mentorship, without which this accomplishment would not have been possible.

I would also like to thank Dr. Jonathan Hopkins, Dr. M. Khalid Jawed, and Dr. Veronica Santos for serving on my committee. Your guidance and expertise have been invaluable throughout this process. I deeply respect each of you as mentors, educators, and researchers, and I am genuinely thankful for the time and effort you have dedicated to supporting my work.

To all the members of RoMeLa, thank you for your incredible support, laughter, and the unforgettable moments we've shared. Each of you has enriched my experience in ways words cannot fully capture. Our discussions, collaborative challenges, late nights, and successes fostered not only a culture of growth and learning but also a sense of family that I will always treasure. The community and camaraderie we built have been essential to my journey. I am sincerely grateful to each of you for your encouragement and for being such an integral part

of my time at the Robotics & Mechanisms Laboratory.

I extend my deepest gratitude to my family, who have been the unwavering pillars of my life. Your love and support have been my foundation throughout this journey. In moments of doubt, your belief in me gave me the strength to move forward, and your encouragement lifted me during challenges. This accomplishment is as much yours as it is mine, a testament to our love and faith. Thank you for being my inspiration and my constant source of motivation.

Finally, I am grateful to my wife, whose strength and love have been the quiet force behind all my achievements. Your unwavering faith in me during the most challenging times has been my guiding light, helping me persevere and succeed. Your patience and encouragement turned obstacles into opportunities, and this journey would not have been the same without your steadfast support. From the depths of my heart, thank you for being my partner, anchor, and greatest source of inspiration.

VITA

2017–2019 M.S. in Mechanical Engineering, UCLA, Los Angeles, California, USA

2011–2017 B.S. in Mechanical Engineering, Sungkyunkwan University,
Seoul, Republic of Korea.

2018–2024 Graduate Student Researcher, UCLA, Los Angeles, California, USA

PUBLICATIONS

[1] **D. Noh**, Y. Liu, F. Rafeedi, H. Nam, K. Gillespie, J. Yi, T. Zhu, Q. Xu, D. Hong, Minimal degree of freedom dual-arm manipulation platform with coupling body joint for diverse cooking tasks. *2020 17th International Conference on Ubiquitous Robots (UR)*, 225-232 (2020).

[2] **D. Noh**, H. Nam, MS Ahn, H. Chae, S. Lee, K. Gillespie, D. Hong, Surface material dataset for robotics applications (SMDRA): A dataset with friction coefficient and RGB-D for surface segmentation. *2020 25th International Conference on Pattern Recognition (ICPR)*, 6275-6281 (2021).

[3] **D. Noh**, H. Nam, K. Gillespie, Y. Liu, D. Hong, YORI: Autonomous Cooking System Utilizing a Modular Robotic Kitchen and a Dual-Arm Proprioceptive Manipulator. *arXiv preprint arXiv:2405.11094* (2024).

- [4] MS Ahn, H. Chae, **D. Noh**, H. Nam, D. Hong, Analysis and noise modeling of the Intel RealSense D435 for mobile robots. *2019 16th International Conference on Ubiquitous Robots (UR)*, 707-711 (2019).
- [5] J. Yi, MS Ahn, H. Chae, H. Nam, **D. Noh**, D. Hong, H. Moon, Task planning with mixed-integer programming for multiple cooking tasks using a dual-arm robot. *2020 17th International Conference on Ubiquitous Robots (UR)*, 29-35 (2020).
- [6] H. Chae, MS Ahn, **D. Noh**, H. Nam, D. Hong, Ballu2: A safe and affordable buoyancy-assisted biped. *Frontiers in Robotics and AI* 8, 730323 (2021).
- [7] J. Yi, T. A. Luong, H. Chae, MS Ahn, **D. Noh**, H. N. Tran, M. Doh, E. Auh, N. Pico, et al., An Online Task-Planning Framework Using Mixed Integer Programming for Multiple Cooking Tasks Using a Dual-Arm Robot. *Applied Sciences* 12 (8), 4018 (2022).

CHAPTER 1

Introduction

1.1 Motivations and Objectives

Robotic manipulators have long served as the backbone of industrial automation, excelling in tasks requiring precision, repeatability, and strength. Historically, their use was confined to structured and isolated environments, where their heavy and rigid design ensured high performance in repetitive operations like welding, assembly, and material handling. However, as society demands more versatile automation, manipulators are being called upon to move beyond industrial applications and into dynamic, unstructured environments, such as homes, healthcare facilities, and restaurants. This shift is driven by the increasing need for robots to perform tasks traditionally reserved for humans, including caregiving, cooking, and collaborative work, especially in anticipation of a declining workforce.

Recent advancements in artificial intelligence and robotics have enabled manipulators to perform more complex tasks with higher autonomy. One significant trend is the integration of manipulators with mobile platforms, resulting in mobile manipulators that combine mobility with dexterity. These systems, which extend the capabilities of traditional manipulators, demonstrate how robotics is addressing real-world challenges in previously inaccessible settings. For example:

- **Drone-mounted manipulators:** Allowing for aerial tasks such as precision agriculture, disaster response, and remote inspection [KCK13, KAS16, RLO18].
- **Wheeled mobile manipulators:** Essential for logistics, performing object-picking

tasks, warehouse automation, and teleoperation in hazardous environments [TSA23, HKS10, THK16].

- **Humanoid robots:** Featuring manipulator arms that mimic human-like tasks, making them ideal for caregiving, domestic chores, and customer service [Ahn23, HO07, GV18, OAS06].

These examples show the potential of mobile manipulators. However, traditional robotic manipulators have inherent limitations due to their high-g geared servo motors. While they perform well in structured environments, their slow joint speed, heavy arm weight, and rigidity render them unsuitable for dynamic, human-centric applications that require adaptability and compliance. Moreover, conventional actuators, optimized for position control, fail to provide the necessary compliance and backdrivability for safe physical interaction. This limitation requires external force and torque sensors, which adds to the system’s complexity. Additionally, many traditional systems operate at lower speeds and have limited torque bandwidth, making them unsuitable for dynamic, dexterous tasks such as throwing, catching, or manipulating fragile objects [HAH08], which leads to the development of parallel manipulators [PG12].

To address these challenges, advancements in robotics have led to the adoption of proprioceptive actuators, particularly in legged robots, where their inherent backdrivability enables lightweight, compliant, and dynamic movements [SWO12, WWS17]. Inspired by this progress, this study explores the adaptation of proprioceptive actuators for robotic manipulators, focusing on leveraging their inherent advantages for impact mitigation and high-bandwidth physical interaction. These actuators offer several key benefits:

- **Backdrivability and Compliance:** Allowing manipulators to yield safely to external forces, enabling intuitive interaction with humans and adaptability to dynamic environments.
- **Lightweight Design:** Reducing the manipulator’s weight decreases inertia, enhancing

control stability and enabling safer environmental interactions while facilitating faster motion capabilities.

- **Force and Torque Control:** Proprioceptive actuators enable precise force and torque control without additional external sensors.
- **High-Speed Operation:** These actuators support faster and more responsive motion, making them suitable for dynamic tasks requiring quick and precise movements.
- **Simplified System Architecture:** By eliminating the need for external sensors, these actuators streamline the overall system, reducing cost and complexity.

Model	Torque (Nm)	RPM (Voltage)	Gear Ratio/Type	Mass (kg)
Dynamixel PH54-200-S500-R	44.7	29 (24V)	501.9:1/Cycloid	0.855
Neuromeka Core200	47.5	25 (48V)	121:1/Harmonic	1.84
Westwood Robotics PB02P	33–67	340.8 (48V)	10:1/Planetary	0.925

Table 1.1: Specifications of Different Robotic Actuators

The distinct features of proprioceptive actuators arise from their low gear ratios, with one of the most remarkable aspects being their high motor speed. As shown in Table 1.1, proprioceptive actuators such as Westwood Robotics PB02P demonstrate speeds up to 13.6 times faster than comparable actuators commonly used in robotics research, such as Dynamixel PH54-200-S500-R, and Neuromeka Core 200.

These advancements redefine the potential of robotic manipulators, transforming them from rigid industrial systems into adaptable tools capable of operating safely and effectively in unstructured, human-centric environments.

This dissertation focuses on advancing a new generation of robotic manipulators optimized for human-centric applications through innovations in control strategies, motion planning, and practical implementations. A crucial aspect involves designing and validating a manipulator that fully utilizes the benefits of proprioceptive actuators, particularly highlighting their fast speed. These efforts aim to achieve dynamic and torque-efficient performance. The effectiveness and adaptability of the proposed manipulator are demonstrated through real-world applications, including autonomous cooking, friction-aware grasping, and dynamic manipulation tasks.

The specific objectives of this research are as follows:

- **Develop a High-Performance Manipulator:** Design a lightweight, high-speed robotic arm with proprioceptive actuators that prioritize safety, compliance, and adaptability.
- **Enhance Motion Planning Techniques:** Develop torque-efficient motion planning algorithms by addressing optimization-based and learning-based approaches, each tailored to minimize peak torque while ensuring dynamic and precise performance.
- **Demonstrate Real-World Applications:** Showcase the manipulator’s capabilities through practical tasks such as cooking, catching objects, and manipulating items using surface friction.
- **Ensure Safety and Robustness:** Develop advanced control strategies that maximize compliance and safety during human-robot interaction, even in unstructured environments.

By achieving these objectives, this work seeks to redefine the role of robotic manipulators, enabling them to address the complex demands of modern human-robot collaboration. These contributions aim to bridge the gap between industrial automation and human-centric applications, paving the way for robotic systems that are not only efficient but also safe, intuitive,

and versatile in their use.

1.2 Background

1.2.1 Manipulator Design

The history of manipulator design can be traced back to the early 20th century, but it was in the 1950s and 1960s that the first industrial robots resembling today’s manipulators were developed. Introduced in 1961, the Unimate was the first industrial robot to replace human labor in die casting. These early manipulators were single-arm systems, primarily hydraulic, and programmed to perform repetitive tasks. Over the years, the technology evolved to include electric actuators, increased degrees of freedom, and more sophisticated control systems, allowing for greater precision and a wider range of applications.

With advancements in safety and sensing technologies, the late 1990s and early 2000s saw the emergence of collaborative robots, or ‘cobots’ [EML19]. Unlike traditional industrial robots, cobots are designed to work alongside humans. They often feature force-torque sensors and advanced algorithms that allow them to understand and adapt to their environment, making them suitable for tasks that require human-robot interaction such as assembly, sorting, and inspection [SSS10].

Efforts to design better collaborative robots for practical use have been ongoing, but developing a versatile robot remains a significant challenge. The most common approach is to create anthropomorphic arms. However, traditional industrial robots have limitations when it comes to versatility. For example, Professor Pieter Abbeel’s Lab at UC Berkeley introduced a Quasi-Direct Drive Manipulator called BLUE in 2019 [GMY19], and Professor Yong-Jae Kim at Koreatech has been continually upgrading his cable-driven dual-arm manipulator since publishing the design paper in 2015 [Kim15]. These novel approaches aim to develop versatile manipulators capable of human-level performance, but many tasks humans perform are still beyond their reach.

The limitations of single-arm manipulators in handling complex tasks have led to the development of dual-arm manipulators. These systems aim to mimic the dexterity and coordination of human arms and hands. Dual-arm manipulators are particularly useful in tasks that require bimanual coordination, such as complex assembly or surgical operations. Companies like ABB and Yaskawa have been pioneers in this space, offering dual-arm robots capable of intricate tasks previously thought to be exclusive to human labor [ZHL20].

The most recent trend in manipulator design is the integration of robotic arms with mobile platforms. These mobile manipulators combine the dexterity of robotic arms with the mobility of platforms like wheeled or legged robots, allowing them to navigate through unstructured environments and perform tasks in various locations. Mobile manipulators are finding applications in warehouses for picking and placing items, in healthcare for patient care, and even in domestic settings for tasks like cleaning and maintenance.

Advances in related technologies have significantly influenced the evolution of manipulator design. Actuators have evolved from hydraulic systems to more efficient and precise electric and pneumatic systems. Advances in materials science have contributed lightweight yet durable materials, allowing for more agile and energy-efficient designs. Control systems have also become increasingly sophisticated, incorporating machine learning algorithms and real-time feedback loops that enable adaptive and intelligent behavior.

The field has come a long way from the rudimentary single-arm systems of the 1960s to today's intelligent, collaborative, and mobile manipulators. As technologies continue to advance, we can expect manipulators to become even more versatile, intelligent, and integral to various aspects of industry and daily life.

1.2.2 Manipulator Control

Control systems serve as the backbone of robotic manipulators, bridging the gap between mechanical design and functional capabilities. Their primary objective is to regulate a robot's

movements and actions precisely and reliably. Over the years, control theories and methodologies have evolved significantly, paralleling advancements in computational power, sensor technology, and mathematical algorithms. This progress has led to various control schemes tailored to specific applications with unique advantages and limitations.

Robotic manipulator control can be broadly categorized into several key areas: position control, torque control, and optimal control methods such as Model Predictive Control (MPC) and Whole-Body Control (WBC).

Position control relies solely on feedback from position error, resulting in a more straightforward control system that often employs precise motors with a high gear ratio [SHV06, SSV09, Cra05]. This approach is straightforward to implement and excels at achieving high accuracy in tasks demanding precise trajectory tracking. A commonly used controller in this domain is the Proportional-Integral-Derivative (PID) controller:

$$u = k_p(q^d - q) + k_d(\dot{q}^d - \dot{q}) + k_i \int (q^d - q) dt, \quad (1.1)$$

Where τ is the control torque, and k_p, k_d, k_i are gains for position, velocity, and integral terms, respectively. Although position control is ideal for tasks that require precise end-effector placement, its simplicity and reliance on high gear ratios render it less suitable for tasks demanding rapid movements or nuanced interaction with the environment.

Torque control strategies focus on explicitly regulating the interaction between the manipulator and its surroundings [Kha87, Mas81, DV88, Hog85]. These methods often utilize six-axis force/torque sensors mounted at the end-effector to measure external forces and moments or torque sensors measuring joint torque on each robot joint. By incorporating detailed robot dynamics models, torque control predicts and manages force-related responses, enabling delicate operations such as precision assembly, polishing, and human-robot collaboration.

The dynamics of a robotic manipulator, including external forces, can be expressed as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + J^\top(q)f_{\text{ext}}, \quad (1.2)$$

The inverse dynamics controller computes the required joint torques to achieve a desired motion by explicitly canceling the system dynamics. Based on the robot’s dynamics equation:

$$u = M(q)\ddot{y} + n(q, \dot{q}) \quad (1.3)$$

where $y = K_p\tilde{q} + K_d\dot{\tilde{q}} + \ddot{q}_d$, $n(q, \dot{q}) = C(q, \dot{q})\dot{q} + F\dot{q} + g(q)$. This controller shows high-precision motion tracking.

The impedance controller regulates the interaction dynamics between the robot and its environment by shaping the relationship between the end-effector position and the applied forces. It could be expressed as:

$$u = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + J^T(K(x^d - x) + D(\dot{x}^d - \dot{x})) + J^T F_{tip}, \quad (1.4)$$

where K and D represent stiffness and damping matrices, x^d and \dot{x}^d are desired end-effector states, and F_{tip} is the external force at the end-effector. By tuning these parameters, the robot can achieve human-like compliance and safe interaction—crucial for tasks in uncertain or dynamic environments.

The increasing complexity of robotic tasks and environments has motivated the development of optimal control frameworks, particularly MPC and WBC. MPC optimizes control inputs over a finite prediction horizon, anticipating future states to ensure smooth and stable performance [MRR00, MC12, Mac02]. The robot’s nonlinear dynamics are often locally linearized and discretized:

$$x_{k+1} = A_k x_k + B_k u_k + r_k, \quad (1.5)$$

where $x_k = [q_k, \dot{q}_k]^T$ is the state vector and u_k the control input. Balancing tracking accuracy, velocity smoothing, and torque minimization, MPC solves an optimization problem:

$$J = \sum_{k=i}^{i+n_p} [(\hat{q}_k - q_k)^T Q_q (\hat{q}_k - q_k) + \dot{q}_k^T Q_v \dot{q}_k + u_k^T Q_u u_k], \quad (1.6)$$

subject to constraints on joint positions, velocities, and torques. By respecting these constraints and anticipating upcoming states, MPC is well-suited for tasks that require ad-

herence to physical and safety limits, such as payload handling, singularity avoidance, and dynamic manipulation.

WBC extends optimal control principles to orchestrate the entire robot, rather than focusing on individual joints or a single end-effector. WBC typically formulates a constrained optimization problem that integrates full-body kinematics, dynamics, and multiple task objectives, often leveraging quadratic programming (QP) techniques. For instance, WBC can be posed as:

$$\begin{aligned} \min_{\dot{q}, \tau, f^{c_i}} \quad & \|J_{\text{task}}(q)\dot{q} - \dot{x}_{\text{task}}^d\|_2^2 + \alpha\|\tau\|_2^2 & (1.7) \\ \text{subject to} \quad & M(q)\dot{\nu} + C(q, \dot{q})\nu + \tau_g(q) = S^T\tau + \sum_i J_{c_i}^T(q)f^{c_i}, \\ & \tau_{\min} \leq \tau \leq \tau_{\max}, \quad f^{c_i} \in \mathcal{C}, \end{aligned}$$

where \dot{q} are joint velocities, τ are joint torques, f^{c_i} are contact forces, and $J_{\text{task}}(q)$ is the task-space Jacobian mapping joint velocities to the desired end-effector velocity \dot{x}_{task}^d . The set \mathcal{C} defines feasible contact wrenches, and S is a selection matrix for actuated degrees of freedom. By formulating a constrained optimization problem that integrates full-body kinematics, dynamics, and multiple task objectives, WBC enables coordinated full-body behaviors [KSP04, RBM13, MBS20].

In summary, the evolution of control strategies for robotic manipulators—from straightforward position control to advanced force/torque and optimal control methods—reflects the growing complexity of modern robotic applications. These advancements have enabled manipulators to achieve unprecedented versatility, robustness, and efficiency, while ensuring safe and effective interaction with dynamic, uncertain environments.

1.2.3 Manipulator Motion Planning

Trajectory generation for robotic manipulators typically involves two steps: finding a geometric path that connects the start and goal points in the configuration space and then

generating a time-parameterized trajectory based on this path. This section explores three widely used approaches: polynomial fitting, time-optimal methods, and minimum jerk trajectories.

Path Generation Methods

To generate a path $P(s)$, two primary techniques are commonly used:

- **Sampling-Based Methods:** Techniques like Rapidly-exploring Random Trees (RRT) and Probabilistic Roadmaps (PRM) are widely used [LP17]. RRT incrementally grows a tree from the start configuration and connects to the goal, while PRM creates a roadmap of sampled points connected by feasible paths [SSV09].
- **Graph-Search Methods:** Algorithms like A* and Dijkstra’s search the configuration space discretized as a grid or graph. These methods are complete and optimal under specific heuristics, making them reliable for structured environments [SSV09].

Trajectory Generation Methods

- **Polynomial Fitting for Trajectory Generation**

Once a path $P(s)$ is determined, a trajectory $q(t)$ can be generated by fitting a polynomial. Polynomials are chosen for their smoothness and simplicity. A general polynomial of degree n is expressed as:

$$q(t) = \sum_{i=0}^n a_i t^i, \quad (1.8)$$

where a_i are coefficients determined by boundary conditions.

- **Cubic Polynomial (n = 3):** Used for cases requiring position and velocity continuity:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3. \quad (1.9)$$

- **Quintic Polynomial ($\mathbf{n} = 5$)**: Provides smoother trajectories by incorporating acceleration continuity:

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5. \quad (1.10)$$

- **Higher-Degree Polynomials ($\mathbf{n} = 7, 9$)**: Allow additional control over jerk or snap, making them ideal for highly dynamic systems [SHV06].

Polynomial fitting can also be formulated as an optimization problem. The coefficients a_i can be computed by solving the following linear system: Polynomial fitting can be formulated as a Quadratic Programming (QP) problem [MK11]. The optimization problem can be expressed as:

$$\begin{aligned} & \text{minimize} && \mathbf{p}^\top \mathbf{Q}\mathbf{p}, \\ & \text{subject to} && \mathbf{A}\mathbf{p} = \mathbf{b}. \end{aligned} \quad (1.11)$$

where \mathbf{p} is the vector of polynomial coefficients, \mathbf{Q} is a known positive semi-definite matrix, and $\mathbf{A}\mathbf{p} = \mathbf{b}$ represents the trajectory constraints, including its derivatives.

- **Time-Optimal Path Parameterization (TOPP)**

Given a predefined geometric path $P(s)$, where s is the arc-length parameter, TOPP focuses on minimizing the traversal time by parameterizing the path with respect to time [Pha14]. The optimization is defined as:

$$\min T = \int_{s_0}^{s_f} \frac{1}{\dot{s}} ds,$$

subject to the following constraints:

$$\begin{aligned} \dot{s} &\geq 0, \\ \ddot{s} &\leq f_{\max}(s, \dot{s}), \\ \ddot{s} &\geq f_{\min}(s, \dot{s}), \end{aligned}$$

where $f_{\max}(s, \dot{s})$ and $f_{\min}(s, \dot{s})$ are functions derived from kinematic constraints, such as velocity and acceleration limits. Traditional TOPP methods primarily handle these

kinematic constraints, but advanced variations may incorporate torque limits to ensure dynamic feasibility along the path [PP18].

- **Time-Optimal Trajectory Planning (TOTP)**

TOTP generalizes the principles of TOPP by incorporating both path parameterization and full trajectory optimization under dynamic constraints. For a given path $P(s)$, the time-optimal trajectory planning problem can be expressed as:

$$\min T = \int_{s_0}^{s_f} \frac{1}{\dot{s}} ds,$$

subject to:

$$\ddot{s} \leq f_{\max}(s, \dot{s}),$$

$$\ddot{s} \geq f_{\min}(s, \dot{s}),$$

$$\tau(s, \dot{s}, \ddot{s}) \leq \tau_{\max},$$

$$\tau(s, \dot{s}, \ddot{s}) \geq \tau_{\min},$$

where $\tau(s, \dot{s}, \ddot{s})$ represents the torque required to follow the trajectory at a given state, and τ_{\max} and τ_{\min} denote the actuator torque limits. By explicitly considering actuator dynamics and external forces, TOTP ensures that the trajectory is both kinematically and dynamically feasible [BDG85, CC00].

- **Minimum Jerk Trajectory**

Minimum jerk trajectories are widely used in applications requiring smooth, human-like motion, such as robotic assistance or rehabilitation. The method minimizes the third derivative of position (jerk) to produce a trajectory that is naturally smooth:

$$\min \int_0^T (\ddot{q}(t))^2 dt,$$

subject to boundary conditions for position, velocity, and acceleration. This leads to a quintic polynomial trajectory of the form:

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5.$$

Table 1.2: Summary of Trajectory Generation Methods

Method	Key Features	Applications
Polynomial Fitting	Smooth, flexible	Industrial robots
Time-Optimal Planning (TOPP, TOPPRA, TOTP)	Minimizes time, handles constraints	High-speed manipulation
Minimum Jerk	Smooth, human-like motion	Assistive robots

The coefficients are computed to satisfy initial and final conditions, ensuring smooth transitions [FH85a].

Table 1.2 summarizes the discussed methods and their key features.

By tailoring the approach to the specific task and constraints, these trajectory generation methods provide effective solutions for robotic manipulators.

1.2.4 Machine Learning for Manipulation

Dexterous and dynamic manipulation hinges on advanced perception and motion planning, which enable robots to interact with their environments intelligently and efficiently. Machine learning (ML) has become a cornerstone in this domain, offering powerful tools to extract meaningful features from multimodal sensor data and bridge the gap between perception and control. This section explores ML-based approaches for perception, focusing on the estimation of physical properties essential for manipulation, and delves into end-to-end learning methods for motion planning, with a particular focus on imitation learning.

Perception for Dexterous and Dynamic Manipulation

Perception is a foundational component of robotic manipulation, enabling machines to understand their environment, recognize objects, and infer the physical properties necessary for effective interaction [SK16]. Unlike traditional industrial settings characterized by highly structured environments and repetitive tasks, modern robotic platforms increasingly operate in dynamic, unstructured, and often unpredictable contexts. This shift demands perceptual

systems capable of interpreting complex sensory inputs and extracting meaningful information to guide control, planning, and decision-making processes [CJP11].

Early approaches to robotic perception focused primarily on vision, leveraging monocular or stereo cameras and model-based techniques to detect and localize known objects [KC02]. Advances in computing power, sensor quality, and algorithmic sophistication have since expanded the perception toolbox. Beyond RGB and RGB-D imagery, contemporary systems incorporate data from a wide array of sensors:

- **Depth and Range Sensing:** Devices such as structured-light sensors, time-of-flight cameras, and lidar provide three-dimensional spatial information which is crucial for understanding object geometry, relative positioning, and environmental topology [CJP11, SK16].
- **Tactile and Force Sensing:** High-resolution tactile sensors and force/torque transducers quantify contact forces, surface textures, and material stiffness [DMV09]. These measurements are essential for delicate grasping, compliant manipulation, and subtle grip adjustments [YZO17].
- **Proprioception:** Joint encoders, torque sensors, and inertial measurement units (IMUs) allow a robot to estimate its internal state, enabling feedback control strategies that improve accuracy, stability, and adaptability [SSV09].
- **Acoustic and Ultrasonic Sensing:** Sound and ultrasound-based measurements can reveal hidden material properties, detect collisions, or identify environmental events [LC23]. Although less common than vision or touch, these modalities offer complementary insights that can enhance perception under conditions where conventional sensors are limited.

Integrating diverse modalities in robotic perception demands sophisticated sensor fusion techniques to address challenges such as varying sensor resolutions, noise, and data rates

[SK16]. Advanced methods, including probabilistic frameworks, machine learning, and optimization, enable the transformation of raw sensory data into actionable information, such as object identities, estimated poses, material properties, and environmental affordances.

Machine learning, particularly deep learning, has driven recent breakthroughs in perception. Models like convolutional neural networks, transformers, and probabilistic generative approaches extract hierarchical features from multimodal datasets [LPK18]. These methods allow robots to infer critical attributes, including grasp points, surface textures, stiffness, and frictional properties [BMA14]. By combining physics-based analytic models with data-driven learning, perception systems achieve improved robustness, interpretability, and adaptability.

End-to-End Learning in Manipulation

End-to-end learning leverages deep neural networks to process raw sensory inputs, such as images or proprioceptive signals, and output control commands. This approach has been successfully applied in various manipulation tasks like pick-and-place, tool use, and deformable object manipulation [LFD16]. Compared to traditional planning algorithms, end-to-end learning, while less structured, demonstrates enhanced adaptability and performance.

Imitation Learning for Motion Planning

Imitation learning is a dominant paradigm in end-to-end motion planning. Imitation Learning enables robots to learn directly from expert demonstrations, bypassing the need for explicit reward design. The following are key Imitation Learning approaches used in manipulation:

- **Behavior Cloning (BC)**: BC treats Imitation Learning as a supervised learning problem. Neural network architectures enhance BC by adapting to various data modalities:
 - **RNN-Based BC**: Recurrent models, such as LSTMs [HS97] and RNN Encoder-Decode [CVG14], capture temporal dependencies, making them suitable for tasks with sequential dynamics, like pouring or folding.

- **CNN-Based BC:** CNNs process spatially structured data, such as images, to perform visually guided manipulation tasks, such as stacking or sorting [HGE17].
- **Diffusion Policies:** Recent advancements have introduced diffusion models for Imitation Learning [Ho22]. These probabilistic models refine action predictions iteratively, handling stochasticity and multimodal behaviors effectively.

Advantages of End-to-End Learning for Motion Planning

End-to-end learning offers the following benefits for manipulation:

- **Direct Optimization:** The network directly maps sensory inputs to task-specific actions, reducing intermediate errors.
- **Adaptability:** These models adapt to new tasks and environments with minimal fine-tuning.
- **Scalability:** They integrate seamlessly with large-scale data collection pipelines for continuous improvement.

Integrating perception models with end-to-end learning for motion planning establishes a cohesive framework for dexterous and dynamic manipulation. By estimating critical physical attributes like friction and stiffness, perception models provide actionable insights that motion planning systems use to execute robust, adaptive actions. This integration enables robots to seamlessly operate in complex, unstructured environments, enhancing their capability to perform tasks with precision and versatility.

1.3 Thesis Organization

This thesis is structured into four main chapters, each addressing a specific aspect of the research, from the development of a novel manipulator to its application in complex tasks and integration of learning-based methods.

Chapter 2 focuses on the development, control, and motion planning of a proprioceptive manipulator. It begins with an overview of the hardware design, highlighting the unique features and capabilities that distinguish it from traditional designs. The chapter then discusses the control strategies used to achieve precise, fast, and robust manipulation, followed by an exploration of motion planning methods designed to fully leverage the manipulator’s advantages. It concludes with a summary of key contributions and technical advancements.

In Chapter 3, the thesis introduces a real-world application called YORI, an autonomous cooking system that combines a modular robotic kitchen with a dual-arm proprioceptive manipulator. The chapter begins with an outline of the overall system design, emphasizing the modularity and adaptability of the kitchen environment. It then examines the implementation of the manipulator’s control and motion planning, focusing on strategies used to address specific challenges. The chapter concludes with demonstrations, showcasing YORI’s ability to autonomously prepare meals in a structured yet dynamic setting.

Chapter 4 presents a novel approach to friction-aware grasping for robotic manipulators. It starts by detailing the preparation and processing of a specialized dataset used to train the proposed grasping framework. The chapter then describes the training model and demonstrates how incorporating friction coefficients enhances grasping efficiency.

Finally, Chapter 6 provides a comprehensive conclusion, summarizing the thesis’s key contributions and reflecting on how the proposed systems and methods address challenges in robotic manipulation. It also outlines future research directions, including the exploration of adaptive video imitation for dynamic manipulation tasks. While preliminary work has investigated using human demonstrations captured in video form to train robots, further

development is needed to realize its full potential. Future efforts will focus on optimizing training processes, enhancing the framework's adaptability, and conducting hardware testing in real-world environments.

Overall, this thesis transitions from foundational system development to advanced applications, offering a cohesive narrative that underscores the integration of hardware design, control strategies, motion planning, and learning-based methods in advancing robotic manipulation.

CHAPTER 2

Development, Motion Planning, and Control of a Proprioceptive Manipulator

2.1 Hardware Design

Despite robotic manipulators' extensive history, we've only seen the incorporation of proprioceptive actuators into these systems in the past decade. The use of such actuators is significant for several reasons. I believe that robotic manipulators gain enhanced versatility with proprioceptive actuators, making them well-suited for assisting humans in real-world, everyday tasks. These actuators enable the robots to operate quickly and safely while offering impact mitigation and precise torque control.

2.1.1 Existing Platforms

In robotic arms, various systems have been engineered to fulfill the requirements of accuracy, speed, and safety. Conventional industrial robots excel in precision and load-bearing capacity but often lack built-in safety mechanisms crucial for human-robot collaboration. Collaborative manipulators have been developed to bridge this safety gap. These robots are equipped with features such as joint torque sensors or force-torque sensors to enhance workplace safety.

While collaborative manipulators are safer to use compared to traditional industrial robots, they still face challenges such as limited speed and payload capacity. The inno-

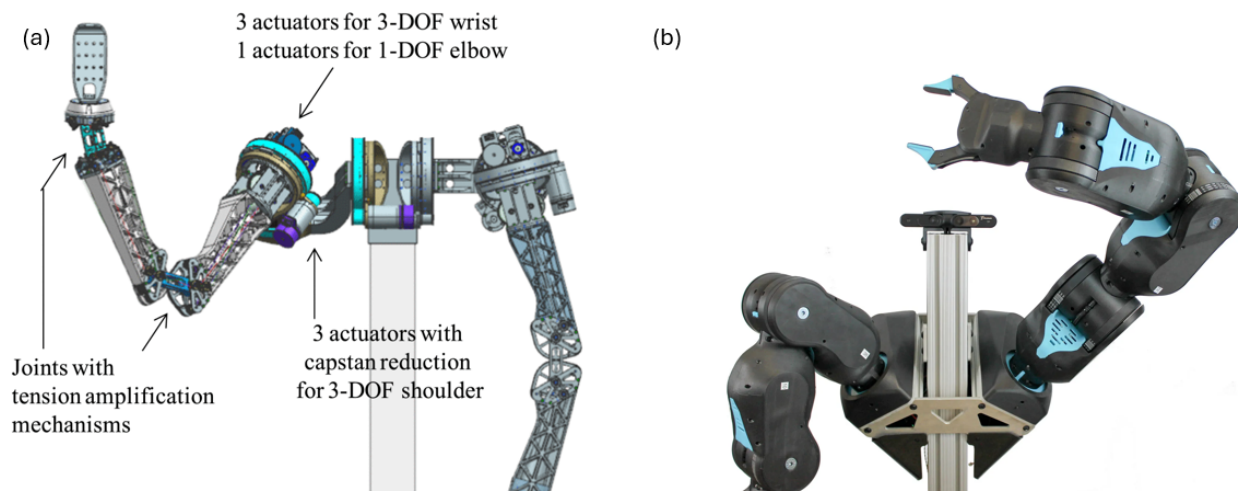


Figure 2.1: (a) LIMS (b) Blue.

vative platform LIMS, shown in (a) of Figure 2.1, addresses this limitation with a 7-DOF robotic arm specifically designed for safe human-robot interaction. [Kim17, SKY18]. LIMS employs a specialized tension amplification mechanism, enabling high joint stiffness without increasing mass or inertia. Compliance with cable-driven power transmission and lightweight and low-inertia arm design ensure inherent safety even during high-speed operations.

UC Berkeley’s BLUE manipulator [GMY19] shown in (b) of Figure 2.1 is another platform worth mentioning. It employs a quasi-direct drive system to achieve cost-effective, compliant robotic manipulation. Like LIMS, BLUE is engineered for safe human interaction, offering an optimal balance between safety and performance.

These next-generation systems distinguish themselves by their back drivability and torque control capability without requiring force-torque sensors, redefining the performance and safety standards for robotic arms.

Our dual-arm manipulator features a torso joint designed to maximize spatial efficiency. Both arms are constructed with a four-bar linkage mechanism, ensuring a lightweight yet rigid structure. This design not only delivers high torque output but also minimizes maintenance requirements. Additionally, hardware specifications such as arm configuration, length,

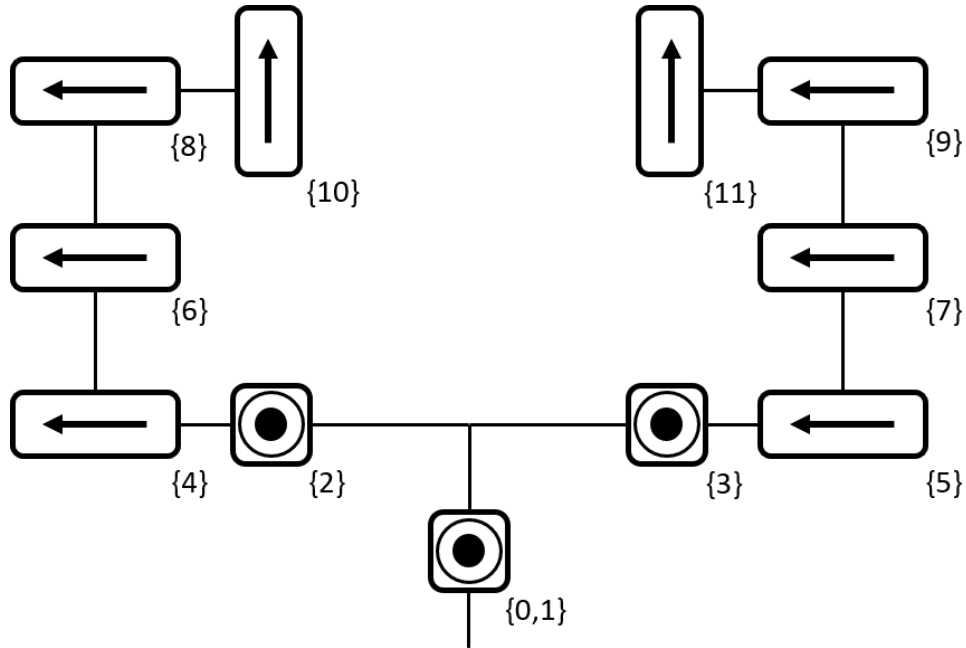


Figure 2.2: Configuration of the dual-arm.

and torque requirements (gear ratios) were finalized after prototype testing using Robotix Dynamixel motors, as shown in Figure 2.4.

The following sections will delve deeper into our platform’s unique attributes compared to others and provide detailed mathematical modeling of our dual-arm system, including frame definitions, kinematics, and dynamics.

2.1.2 Design Overview

The dual-arm manipulation platform features two arms connected by a shoulder link. Each arm of the dual-arm manipulator has five Degrees of Freedom (DoFs). The first two DoFs are located at the shoulder and are orthogonal to each other. The third DoF, another pitch, is located at the elbow, followed by two additional DoFs at the wrist. Actuators at each joint are serially interconnected by arm linkages. This five-DoF design not only reduces the arm’s weight and inertia but also prevents system instability caused by singularities.

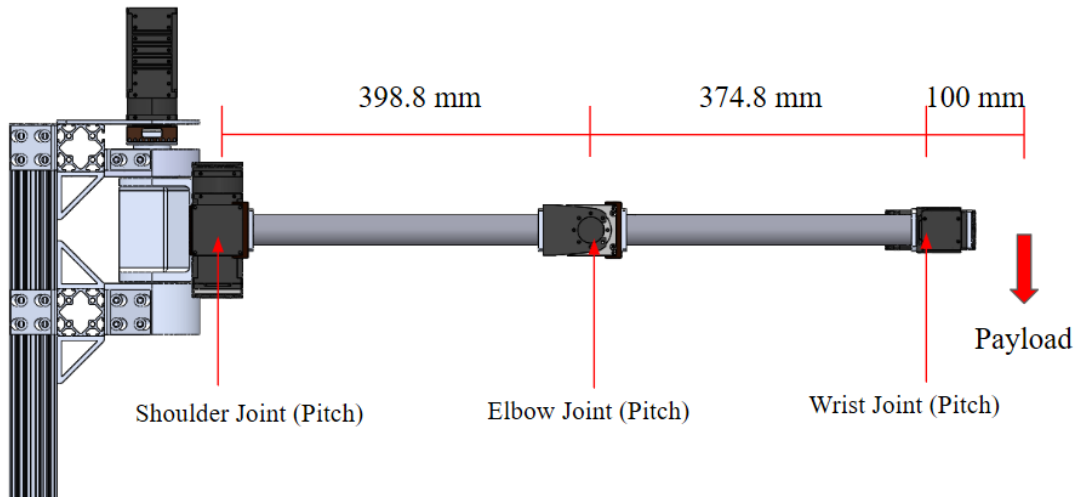


Figure 2.3: Pose with fully extended arm in horizontal direction.

The robot incorporates a rotational joint (Yaw) at the connection between the torso and shoulder, enabling the entire shoulder assembly to rotate. To create a dual-arm manipulator designed for human tasks, the upper arm and forearm links each measure 304.8 mm (1 foot) in length, while the shoulder link is 450 mm long. The total height of the manipulator platform is approximately 1450 mm. This configuration provides a substantial and overlapping reachable workspace for both arms, enabling efficient dual-arm manipulation. Additionally, an extra degree of freedom at the platform’s waist expands the reachable workspace, covering most of the structured cooking environment. Carbon fiber was chosen as the material for the arm and shoulder links due to its high stiffness-to-weight ratio.

2.1.3 Prototype: Dual-Arm Manipulator With High Gear Ratio Servo Motor

The torque requirements for each actuator are determined using the static moment equilibrium method, which calculates the maximum torque needed when the arm is fully extended. This analysis considers a standard 2 kg payload for cooking tasks, positioned 100mm from the wrist joint. The arm segment lengths are 398.8mm from shoulder to elbow and 374.8 mm from elbow to wrist, as depicted in Figure 2.3. The actuators at the shoulder and elbow are

Table 2.1: Torque Requirement for Each Pitch Actuator

Actuator	Shoulder Pitch	Elbow Pitch	Wrist Pitch
Required Torque (N·m)	37.21	17.65	2.94

assumed to weigh 1kg each, while the wrist actuator is estimated at 0.5kg. After calculating the minimum torque requirements, a safety factor of 1.5 is applied to account for inertial effects, resulting in the final torque specifications for each pitch actuator, summarized in Table 2.1.

A prototype was assembled to validate the design’s practicality and workspace. The preceding torque analysis guided the selection of actuators. We chose Dynamixel Pro Plus actuators from Robotis, which is known for their robust performance. The Dynamixel PH42-020-S300-R, with a continuous torque of 5.1 N·m and a weight of 340 grams, was found to be adequate for the pitch actuator in the wrist joint. For the pitch actuators in the shoulder and elbow joints, the Dynamixel PH54-200-S500-R was chosen, providing a significant continuous torque of 44.7 N·m and a weight of 855 grams. For ease of installation and control, this actuator model was also used for the waist and the yaw actuator at the shoulder joint. The Dynamixel PH42-020-S300-R was employed for the roll actuator at the wrist joint. With these actuators, the weight of a single arm is 4.39 kg.

The successfully fabricated prototype of the dual-arm manipulation platform is illustrated in Figure 2.4. All parts were manufactured in the lab besides Dynamixel-related parts.



Figure 2.4: Prototype of the dual-arm manipulator.

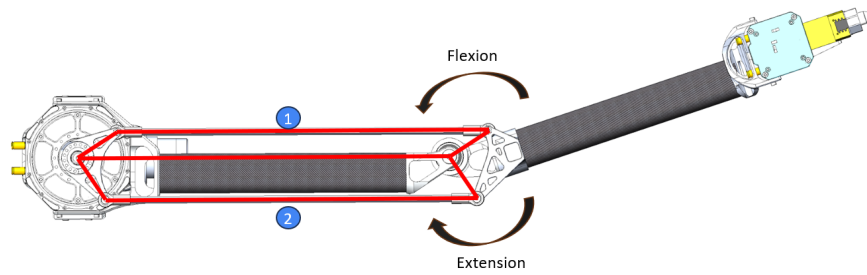


Figure 2.5: 4-Bar linkage.

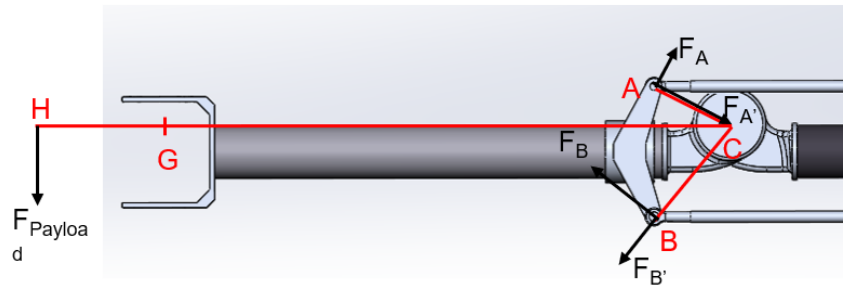


Figure 2.6: Forearm.

2.1.4 Dual-Arm Manipulator Equipped Proprioceptive Actuators

After testing the prototype, we began the construction of a dual-arm manipulator equipped with proprioceptive actuators. We utilized existing BEAR actuators from our lab's spin-off company, Westwood Robotics, and customized a specific type of BEAR actuator to achieve higher torque. To minimize the inertia of the arm, we selected a 4-bar linkage power transmission method to actuate the elbow joint, as depicted in Figure 2.5. Based on this mechanism, we calculated the required torque as follows:

The torque requirements for the robot's wrist, elbow, and shoulder pitch joints were calculated using static equilibrium principles, considering the arm's geometry and payload distribution.

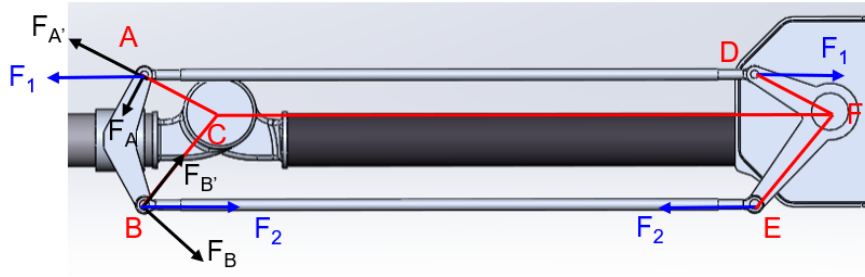


Figure 2.7: Upperarm.

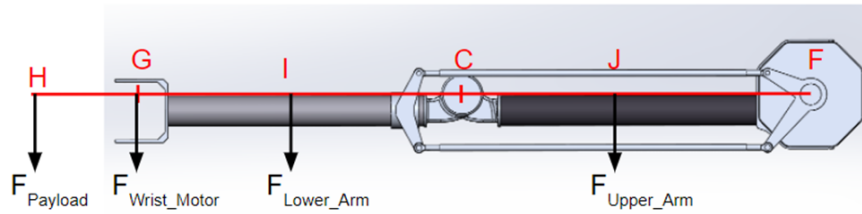


Figure 2.8: Entire arm.

Wrist Torque Calculation

The torque required at the wrist motor is calculated as:

$$T_{\text{wrist_motor}} = F_{\text{payload}} \cdot HG = 0.15 \cdot F_{\text{payload}} \text{ (N} \cdot \text{m)}, \quad (2.1)$$

where $HG = 150$ mm, as shown in Figure 2.6. For a maximum payload of 3 kg, the required torque at the wrist is:

$$T_{\text{wrist_motor}} = 0.15 \cdot 30 = 4.5 \text{ N} \cdot \text{m}.$$

Elbow Torque Calculation

The torque required at the elbow joint was calculated using the following equations of static equilibrium:

$$F_A \cdot AC = F_B \cdot BC = 0.5 \cdot F_{\text{payload}} \cdot HC, \quad (2.2)$$

where:

$$F_A = 5 \cdot F_{\text{payload}}, \quad F_B = 3.6 \cdot F_{\text{payload}}, \quad F_1 = 11.3 \cdot F_{\text{payload}}, \quad F_2 = 4.6 \cdot F_{\text{payload}}, \quad (2.3)$$

and the geometric parameters are defined as:

$$\angle ACG = \angle DFC = 26.32^\circ, \quad \angle BCG = \angle EFC = 50.83^\circ,$$

$$AC = DF = 51.88 \text{ mm}, \quad BC = EF = 73.61 \text{ mm},$$

$$CG = 374.8 \text{ mm}, \quad HG = 150 \text{ mm},$$

as illustrated in Figure 2.7. The torque at the elbow joint is computed as:

$$T_{\text{elbow}} = F_1 \cdot DF + F_2 \cdot EF = 0.52 \cdot F_{\text{payload}} \text{ (N} \cdot \text{m)}. \quad (2.4)$$

Assuming $F_{\text{payload}} = 30 \text{ N}$ (payload) and 5 N (weight of the lower arm and wrist motors), the total torque required is:

$$T_{\text{elbow}} = 0.52 \cdot 35 = 18.2 \text{ N} \cdot \text{m}.$$

Shoulder Torque Calculation

The torque required at the shoulder pitch joint is determined as:

$$T_{\text{shoulder}} = F_{\text{payload}} \cdot HF + F_{\text{wrist}} \cdot GF + F_{\text{lowerarm}} \cdot IF + F_{\text{upperarm}} \cdot JF, \quad (2.5)$$

where:

$$F_{\text{payload}} = 30 \text{ N},$$

$$F_{\text{wrist}} = 2.5 \text{ N} \cdot 2 = 5 \text{ N},$$

$$F_{\text{lowerarm}} = 3 \text{ N},$$

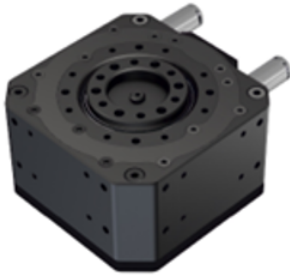
$$F_{\text{upperarm}} = 3 \text{ N}.$$

The geometric parameters are defined as:

$$CF = 398.88 \text{ mm}, \quad CG = 374.8 \text{ mm}, \quad HG = 150 \text{ mm},$$

as shown in Figure 2.8. Substituting these values, the torque required at the shoulder is:

$$T_{\text{shoulder}} = 33.9 \text{ N} \cdot \text{m}.$$



Koala BEAR

Dimensions: 63.5x62x37mm Stall Torque - 15 sec: 4.2Nm
Weight: 250g Stall Torque - 1.5 sec: 10.5Nm
Voltage: 9~33.6V (3~8S) Speed Constant: 27.3RPM/V



Panda BEAR

Dimensions: 113x113x49.7mm Stall Torque - 15 sec: 16.8Nm
Weight: 685g Stall Torque - 1.5 sec: 33.5Nm
Voltage: 9~50.4V (3~12S) Speed Constant: 14.3RPM/V

Panda BEAR Plus

Dimensions: 113x113x49.7mm Stall Torque - 15 sec: 33Nm
Weight: 925g Stall Torque - 1.5 sec: 67Nm
Voltage: 9~50.4V (3~12S) Speed Constant: 7.1RPM/V

Figure 2.9: BEAR series.

Based on these torque calculations, the types of actuators were selected, as illustrated in Figure 2.9. We collaborated with Westwood Robotics to develop the Panda BEAR Plus, which offers higher torque while maintaining a similar size. The completed dual-arm manipulator is depicted in Figure 2.10.

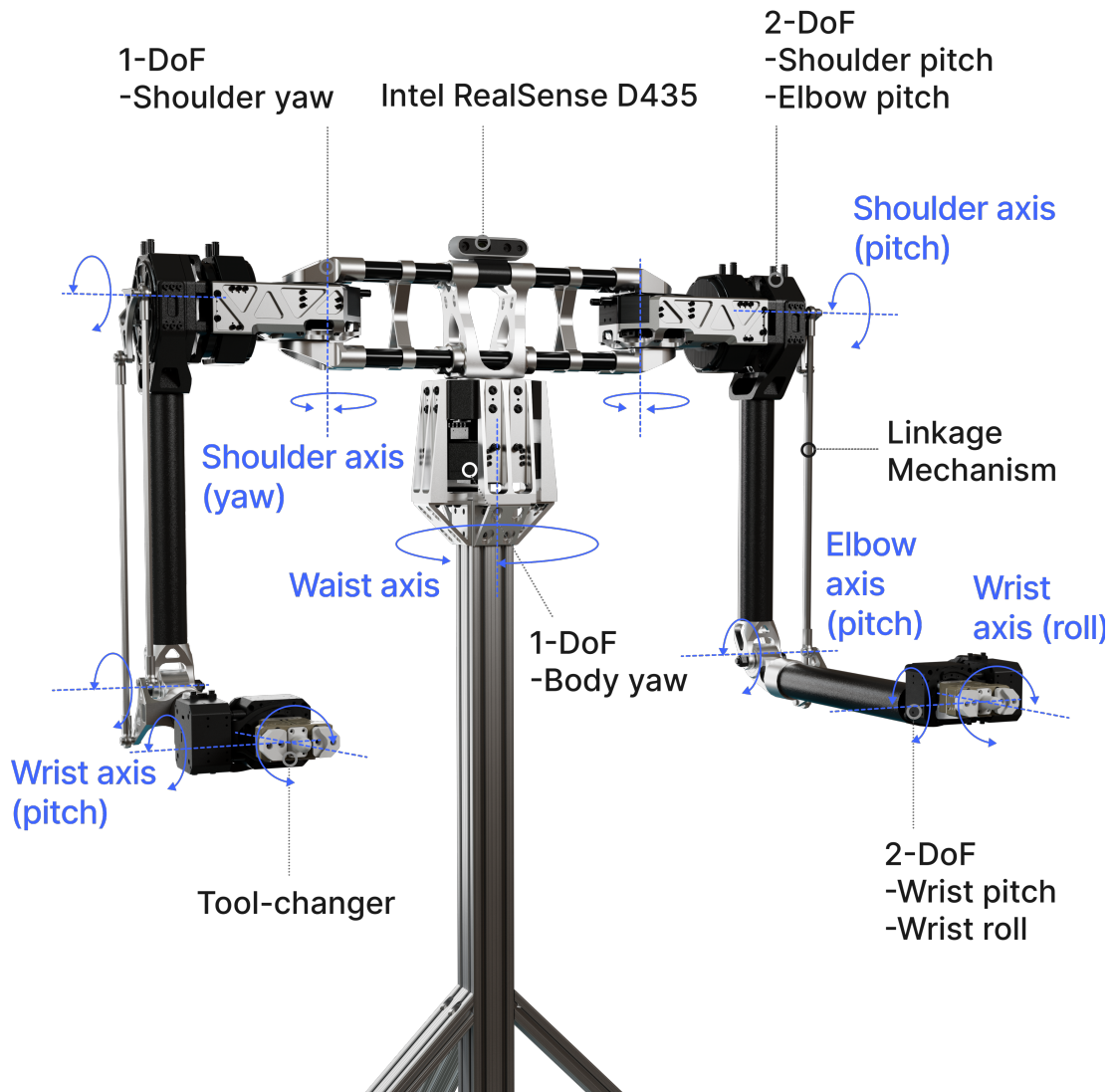


Figure 2.10: Dual-arm manipulation platform equipped proprioceptive actuators.

2.2 Modeling

2.2.1 Definition

Before undertaking any kinematic or dynamic analyses, it's essential to define the coordinate axes for both the links and the joints. Our dual-arm manipulator is a multi-rigid body system that can be represented using a rigid-body tree structure. Starting from its body frame, the manipulator consists of two rigid-body chains. The names of the joints and their offsets from their parent links are detailed in Table 2.2. The coordinate axes are illustrated in Figure 2.12 and Figure 2.11.

	Joint	ID	Coordinate
Left	l1	2	[0, 0.210, 0]
	l2	3	[0, 0.360, 0]
	l3	4	[0, 0.360, -0.4]
	l4	5	[0.375, 0.360, -0.4]
	l5	6	[0.45975, 0.360, -0.4]
Right	r1	7	[0, -0.210, 0]
	r2	8	[0, -0.360, 0]
	r3	9	[0, -0.360, -0.4]
	r4	10	[0.375, -0.360, -0.4]
	r5	11	[0.45975, -0.360, -0.4]
Body	body	1	[0, 0, 0]

Table 2.2: Definition of the Coordinate Axes

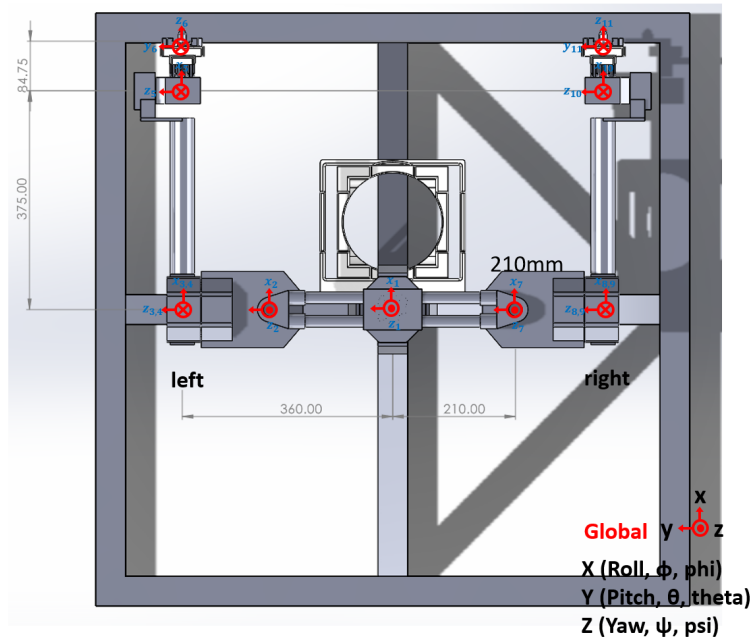


Figure 2.11: Coordinate axis diagram: left view.

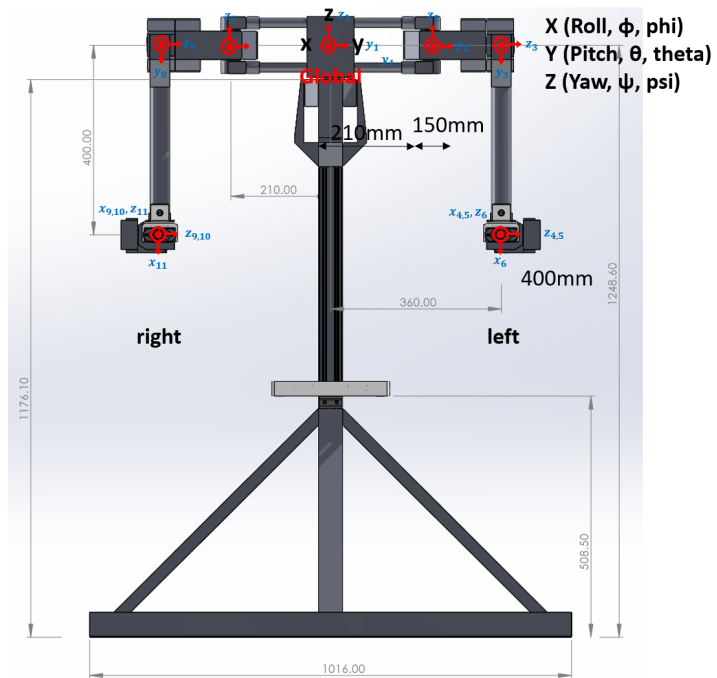


Figure 2.12: Coordinate axis diagram: front view.

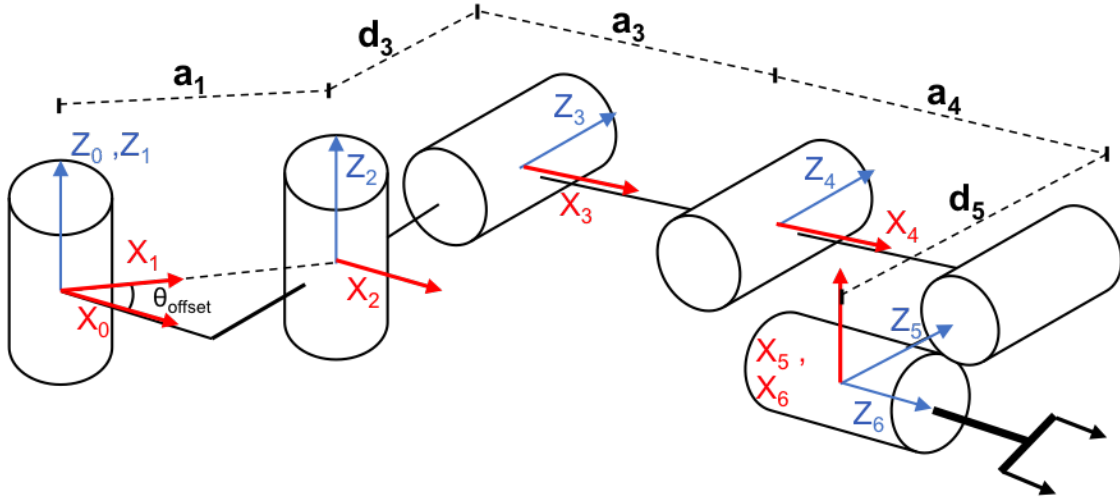


Figure 2.13: Left arm frame assignment for DH parameters.

2.2.2 Kinematics

2.2.2.1 Forward Kinematics

The forward kinematics of a robot calculates the position and orientation of its end-effector frame relative to its base frame based on its joint angles. Here, the Denavit-Hartenberg (D-H) representation is used to represent the forward kinematics. The forward kinematics of the left arm is presented in this section. As the right arm has a mirrored configuration, the forward kinematics of the right arm can be done similarly.

The assigned orientations of all joint frames are shown below in Figure 2.13. It should be noted that the base frame is defined at the robot's torso and that the last two joints are combined at the intersection point of those two joint axes. The linkage frames are established based on D-H rules, and the corresponding D-H parameters are shown in Table 2.4. Note that $\tilde{\theta}_1 = \theta_1 + \theta_{offset}$ and $\tilde{\theta}_2 = \theta_2 - \theta_{offset}$. Also consider that d_5 in Table 2.3 is -0.0840m because it is the distance from \mathbf{X}_4 to \mathbf{X}_5 measured along \mathbf{Z}_5 .

Symbol	Values
a_1	0.2886m
a_3	0.3988m
a_4	0.3748m
d_3	0.1020m
d_5	-0.0840m
θ_{offset}	75.96°

Table 2.3: Constant Parameters for Frames

The homogeneous transformation matrix between adjacent link frames is defined as:

$${}^{i-1}T_i = \begin{bmatrix} c_i & -s_i & 0 & a_{i-1} \\ s_i c\alpha_{i-1} & c_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s_i s\alpha_{i-1} & c_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

where c_i and s_i represent $\cos \theta_i$ and $\sin \theta_i$, respectively.

The Cartesian position and orientation of the end-effector relative to the base frame is computed by 0T_6 , which is a function of all joint variables. 0T_6 is computed and presented below:

$${}^0T_6 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (2.7)$$

$${}^0T_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

where $r_{11} = s_{12}s_6 + c_{12}s_{345}c_6$, $r_{12} = s_{12}c_6 - c_{12}s_{345}s_6$, $r_{13} = c_{12}c_{345}$, $r_{21} = s_{12}s_{345}c_6 - c_{12}s_6$, $r_{22} = -s_{12}s_{345}s_6 - c_{12}c_6$, $r_{23} = s_{12}c_{345}$, $r_{31} = c_{345}c_6$, $r_{32} = -c_{345}s_6$, $r_{33} = -s_{345}$, $P_x =$

Link	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	$\tilde{\theta}_1$
2	a_1	0	0	$\tilde{\theta}_2$
3	0	$-\pi/2$	d_3	θ_3
4	a_3	0	0	θ_4
5	a_4	0	d_5	$\theta_5 - \pi/2$
6	0	$-\pi/2$	0	$-\pi/2$

Table 2.4: DH Parameters of the Left Arm

$$a_1c_1 + a_3c_{12}c_3 + a_4c_{12}c_{34} - d_3s_{12} - d_5s_{12}, P_y = a_1s_1 + a_3s_{12}c_3 + a_4s_{12}c_{34} + d_3c_{12} + d_5c_{12},$$

$$P_z = -a_3s_3 - a_4s_{34}.$$

2.2.2.2 Inverse Kinematics

Inverse kinematics is used to control the movement of a robotic arm through the calculation of the required joint angles needed for the end-effector frame to reach a chosen position and orientation with respect to its base frame. For 6 Dof arm, which includes body joint, the inverse kinematic solution can be obtained analytically. Below is how we get the analytical solution of inverse kinematics. This solution provides all the possible solutions for inverse kinematics. This enables us to choose the optimal path of the manipulator based on its initial state and the surrounding environment.

To solve the inverse kinematics, we will consider a given orientation and position of the end-effector B_P and B_R respectively. The goal position can be represented as:

$${}^B_E T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^B_E R & {}^B_E P \\ 0 & 1 \end{bmatrix} \quad (2.9)$$

To get the inverse kinematics, the following equation will be solved,

$${}^B_E T = {}^0_6 T \quad (2.10)$$

First, the dependence on $\tilde{\theta}_1$ will be placed on the left-hand side of the Equation 2.10 as follows,

$$[{}^0_1 T]^{-1} {}^B_E T = {}^1_6 T \quad (2.11)$$

Next, the following equations can be obtained by using Equation 2.11.

$$r_{31} = c_6 c_{345}, r_{32} = -s_6 c_{345}, r_{33} = -s_{345} \quad (2.12)$$

Thus, θ_6 and the combined angle $\theta_{345} = \theta_3 + \theta_4 + \theta_5$ can be decided as Equation 2.13 and Equation 2.14.

$$\theta_3 + \theta_4 + \theta_5 = \text{Atan2} \left(-r_{33}, \mp \sqrt{r_{31}^2 + r_{32}^2} \right) \quad (2.13)$$

$$\theta_6 = \text{Atan2} (-r_{32} / \cos \theta_{345}, r_{31} / \cos \theta_{345}) \quad (2.14)$$

Similarly, by placing the dependence on both $\tilde{\theta}_1$ and $\tilde{\theta}_2$ on the left-hand side of Equation 2.10, we can solve for both $\tilde{\theta}_1$ and $\tilde{\theta}_2$

$$[{}^1_2 T]^{-1} [{}^0_1 T]^{-1} {}^B_E T = {}^2_6 T \quad (2.15)$$

Using Equation 2.15, the following can be obtained.

$$\tilde{s}_{12} = \frac{c_6 s_{345} r_{22} + s_6 s_{345} r_{21}}{r_{11} r_{22} - r_{21} r_{21}} = a \tilde{c}_{12} = \frac{-c_6 c_{345} r_{12} + s_6 s_{345} r_{22}}{r_{11} r_{22} - r_{12} r_{22}} = b \quad (2.16)$$

$$\tilde{\theta}_1 + \tilde{\theta}_2 = \theta_1 + \theta_2 = \text{Atan2}(a, b) \quad (2.17)$$

Since the expression $\theta_1 + \theta_2$ is now known, it can be used to obtain θ_2 using,

$$\tilde{s}_2 = \frac{d_5 + d_3 + P_x s_{12} - P_y c_{12}}{a_1} = c \quad (2.18)$$

Thus,

$$\tilde{\theta}_2 = \text{Atan2}\left(c, \pm\sqrt{1-c^2}\right) \quad (2.19)$$

$$\theta_2 = \text{Atan2}\left(c, \pm\sqrt{1-c^2}\right) + \theta_{offset} \quad (2.20)$$

and,

$$\theta_1 = \text{Atan2}(a, b) - \text{Atan2}\left(c, \pm\sqrt{1-c^2}\right) - \theta_{offset} \quad (2.21)$$

Again using Equation 2.15, we can write the following equations.

$$P_x c_{12} + P_y s_{12} - a_1 \tilde{c}_2 = a_4 c_{34} + a_3 c_3 \quad (2.22)$$

$$-P_x s_{12} + P_y c_{12} + a_1 \tilde{s}_2 = a_4 c_{34} + a_3 c_3 \quad (2.23)$$

$$P_z s_{12} = -a_4 s_{34} - a_3 s_3 \quad (2.24)$$

By squaring the previous three equations and summing them, we can write isolate $\cos(\theta_4)$ and write θ_4

$$\theta_4 = \text{Atan2}\left(\pm\sqrt{1-c_4^2}, c_4\right) \quad (2.25)$$

Finally, we place the dependence on $\tilde{\theta}_1$, $\tilde{\theta}_2$ and θ_3 on the left side of Equation 2.10

$$\begin{bmatrix} 2T \\ 3T \end{bmatrix}^{-1} \begin{bmatrix} 1T \\ 2T \end{bmatrix}^{-1} \begin{bmatrix} 0T \\ 1T \end{bmatrix}^{-1} \begin{matrix} B \\ E \end{matrix} T = \begin{matrix} 3T \\ 6T \end{matrix} \quad (2.26)$$

Using Equation 2.26, we can come up with the following expressions.

$$d \cos \theta_3 + e \sin \theta_3 = f \quad (2.27)$$

$$g \cos \theta_3 + h \sin \theta_3 = i \quad (2.28)$$

Where,

$$d = P_x c_{12} + P_y s_{12} - a_1 \tilde{c}_2 e = -P_z f = a_3 + a_4 c_4 g = -P_z h = -P_x s_{12} + P_y c_{12} + a_1 \tilde{s}_2 i = a_4 s_4 \quad (2.29)$$

And then, solve for θ_3

$$\theta_3 = \text{Atan2}(di - df, fh - ei) \quad (2.30)$$

Lastly, we can calculate θ_5 using Equation 2.12, Equation 2.25, and Equation 2.30.

$$\theta_5 = \text{Atan2}\left(-r_{33}, \pm \sqrt{r_{31}^2 + r_{32}^2}\right) - \theta_4 - \theta_3 \quad (2.31)$$

Numerical approaches are widely employed for solving inverse kinematics, particularly for robotic arms with limited degrees of freedom (e.g., 5-DOF arms, excluding the base joint). Among these, the Weighted Damped Least Squares (WDLS) method is a robust and effective technique for handling joint limits while ensuring numerical stability. Algorithm 1 presents the pseudocode for the implementation of WDLS in our framework.

The weight matrix W in the WDLS method is critical for addressing joint limits. It is defined as follows:

$$W = \text{diag}\left(\frac{1}{(q - q_{\min})^2 + \epsilon} + \frac{1}{(q_{\max} - q)^2 + \epsilon}\right), \quad (2.32)$$

where q_{\min} and q_{\max} represent the lower and upper joint limits, respectively, and ϵ is a small constant added to ensure numerical stability. By applying this weighting strategy, joint movements are naturally constrained within their permissible ranges while maintaining smooth and stable updates.

Pseudocode 1 Weighted Damped Least Squares (WDLS) for Inverse Kinematics

Require: $E, T_{\text{dBE}}, q_{\text{initial}}, W(q), n_{\text{iter}}, dt, \epsilon, \lambda$

```
1:  $q \leftarrow q_{\text{initial}}$  {Initialize joint angles}
2:  $i \leftarrow 0$ 
3: while  $i < n_{\text{iter}}$  do
4:    $T_{\text{BE}} \leftarrow FK(q, E)$  {Compute forward kinematics}
5:    $\text{err} \leftarrow \log_6(T_{\text{BE}}, T_{\text{dBE}})$  {Calculate pose error}
6:   if  $\|\text{err}\| < \epsilon$  then
7:      $\text{success\_stat} \leftarrow \text{success}$ 
8:     break
9:   end if
10:   $J \leftarrow \text{get\_jacobian}(q, E)$  {Compute Jacobian matrix}
11:   $W \leftarrow \text{compute\_weight}(q)$  {Compute weight matrix based on joint limits}
12:   $\dot{q} \leftarrow WJ^T(JWJ^T + \lambda I)^{-1}\text{err}$  {WDLS update}
13:   $q \leftarrow q + \dot{q} \cdot dt$  {Update joint angles using Euler integration}
14:   $i \leftarrow i + 1$ 
15: end while
16: if  $i \geq n_{\text{iter}}$  then
17:    $\text{success\_stat} \leftarrow \text{fail}$ 
18: end if
19: return  $q, \text{success\_stat}$ 
```

2.2.3 Dynamics

Dynamics gives us insight into why things are moving as they are, under the influence of forces and torques. Since we expect high-speed dynamic motion and accurate position tracking for our manipulator platform, controlling the manipulator using only its kinematics is a challenge. In addition, as it has a coupled body joint that is connected to two relatively heavy arms that could have a large inertia when it is straightened, dynamics must be considered when it is controlled. We can define the well-known equations of motion or the manipulator equations as follows.

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + g(\theta) + J^T(\theta)F_{\text{tip}} \quad (2.33)$$

where $M(\theta)$ is the $n \times n$ symmetric positive-definite mass matrix, $c(\theta, \dot{\theta})$ is the $n \times n$ Coriolis matrix, $g(\theta)$ are gravitational forces, F_{tip} is the wrench applied by the manipulator to the environment.

2.3 Control Strategies

The following pseudo-code implements inverse dynamics and impedance control for a dual-arm manipulator with a shared base joint. Each arm is independently controlled, and a coordination mechanism handles the shared joint dynamics.

2.3.1 Inverse Dynamics Controller Implementation

Pseudocode 2 Dual-Arm Inverse Dynamics

Require: $\mathbf{H}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, $\mathbf{g}(\mathbf{q})$, \mathbf{J} , $\dot{\mathbf{J}}$, \mathbf{q}_d , $\dot{\mathbf{q}}_d$, $\ddot{\mathbf{q}}_d$, \mathbf{F}_{ext}

- 1: $\boldsymbol{\tau}_{\text{torso}} \leftarrow \mathbf{0}$ {Initialize shared torso torque}
 - 2: **for** each arm (left, right) **do**
 - 3: $\ddot{\mathbf{q}}_d \leftarrow \mathbf{J}^{-1}(\ddot{\mathbf{x}}_d - \dot{\mathbf{J}}\dot{\mathbf{q}})$ {Compute desired joint acceleration}
 - 4: $\boldsymbol{\tau} \leftarrow \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{J}^T \mathbf{F}_{\text{ext}}$ {Compute joint torques}
 - 5: Accumulate $\boldsymbol{\tau}_{\text{torso}}$ using a weight factor α
 - 6: $\boldsymbol{\tau}_{\text{torso}} \leftarrow \boldsymbol{\tau}_{\text{torso}} + \alpha \boldsymbol{\tau}$
 - 7: **end for**
 - 8: Normalize $\boldsymbol{\tau}_{\text{torso}}$ {Distribute shared joint torque}
 - 9: **return** $\boldsymbol{\tau}_{\text{left}}, \boldsymbol{\tau}_{\text{right}}, \boldsymbol{\tau}_{\text{torso}}$
-

A repeatability test based on ISO 9283 was conducted using the Inverse Dynamics controller to evaluate the manipulator's performance. The end-effector was moved repeatedly along four designated points, and its position was tracked using a motion capture system (Vicon Vero). The testing trajectory is shown in Figure 2.14, and the average deviation was measured as 0.493 mm (see Table 2.5). This result is comparable to those of LIMS and KUKA iiwa (see Table 2.6), demonstrating that the proprioceptive manipulator offers sufficient position accuracy.

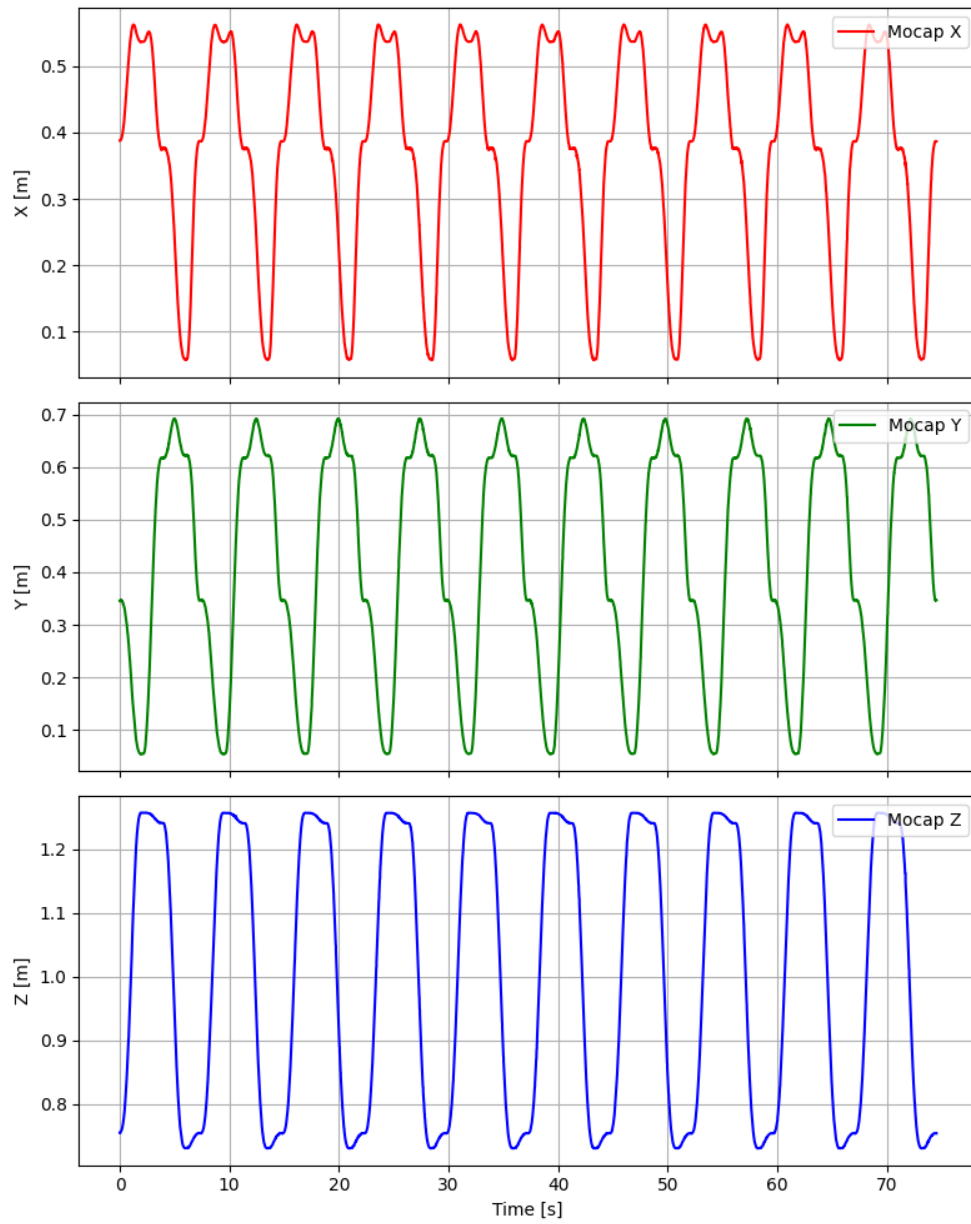


Figure 2.14: Repeatability test results based on motion capture tracking.

Point	Deviation (mm)
P1	0.585
P2	0.264
P3	0.681
P4	0.443
Average	0.493

Table 2.5: Deviation Measurements Recorded at Designated Points

Manipulator	Deviation (mm)
LIMS	0.425
KUKA iiwa	0.100

Table 2.6: Deviation Measurements for Different Manipulators

Next, to assess tracking performance, the manipulator was maneuvered through a series of positions, labeled 1 through 9, as illustrated in Figure 2.15. Position controller and Inverse Dynamics controller were used for this evaluation. The actual x, y, and z coordinates of the end-effector were compared to the reference trajectory. Tests were conducted at two speeds (1 rad/s, 3 rad/s) and with varying payloads (0 kg, 1 kg, and 2 kg).

The position controller showed increasing errors with higher speeds and payloads, even after fine-tuning. In contrast, the Inverse Dynamics controller demonstrated stable and accurate tracking under all conditions.

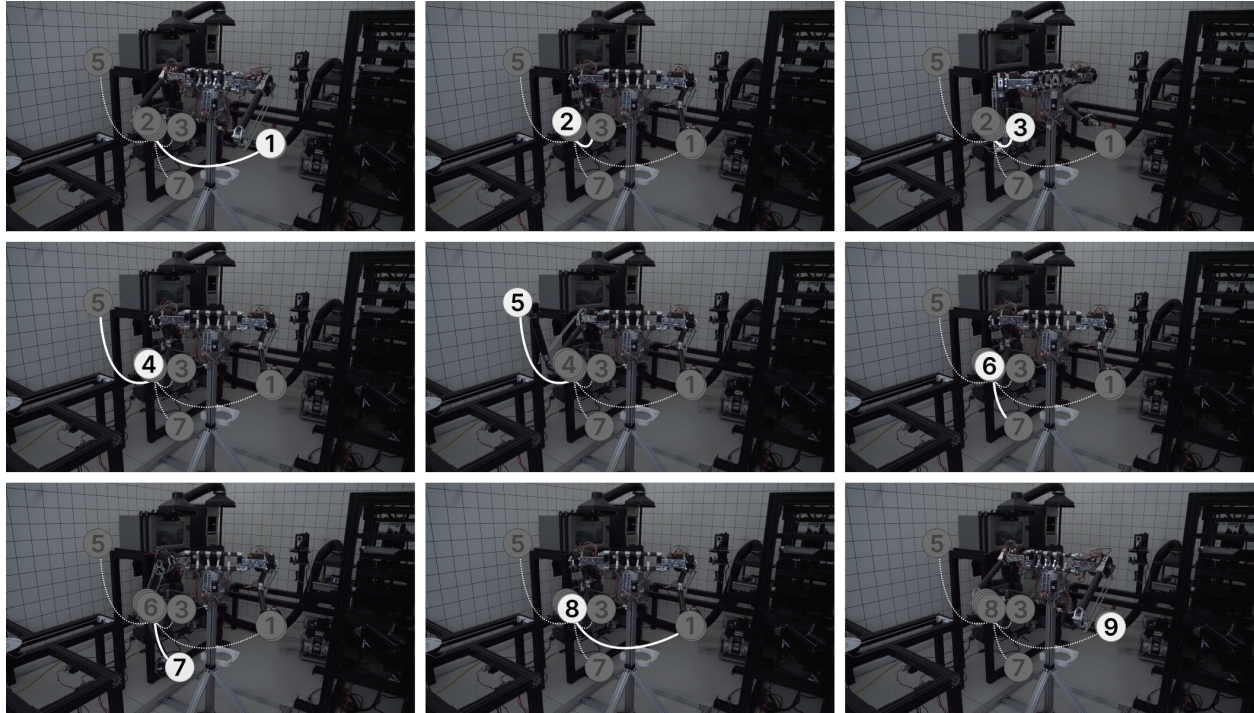


Figure 2.15: Tracking performance experiment sequence.

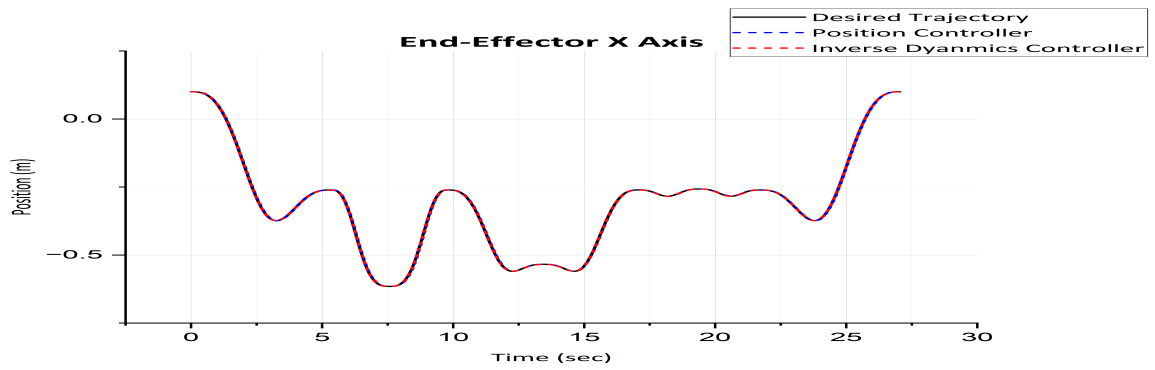


Figure 2.16: Result: end-effector x-axis / 1 rad/s / 0kg.

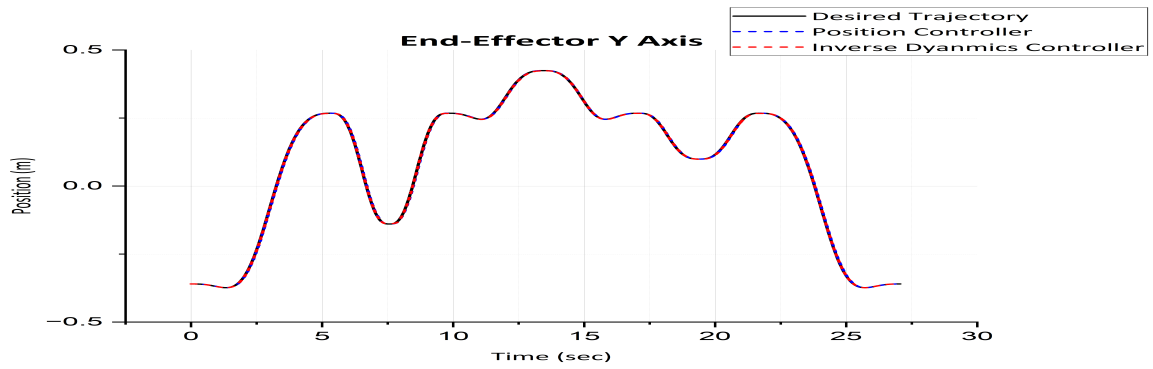


Figure 2.17: Result: end-effector y axis / 1 rad/s / 0kg.

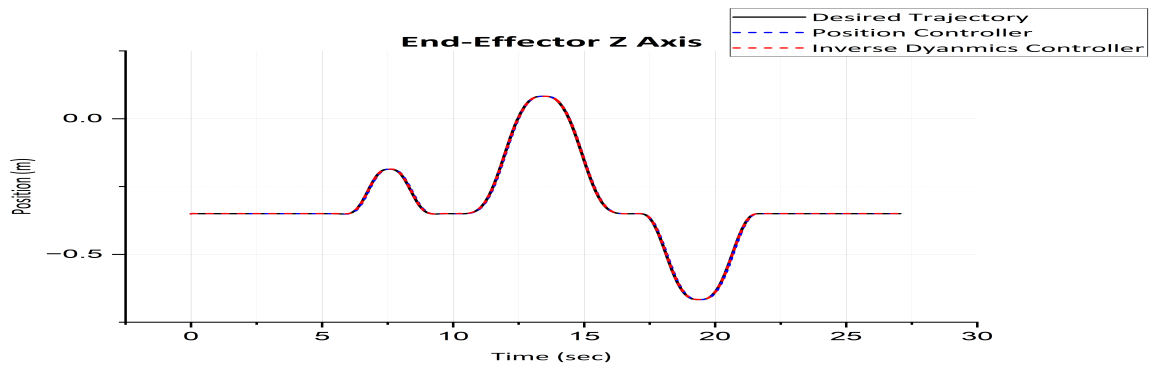


Figure 2.18: Result: end-effector z-axis / 1 rad/s / 0kg.

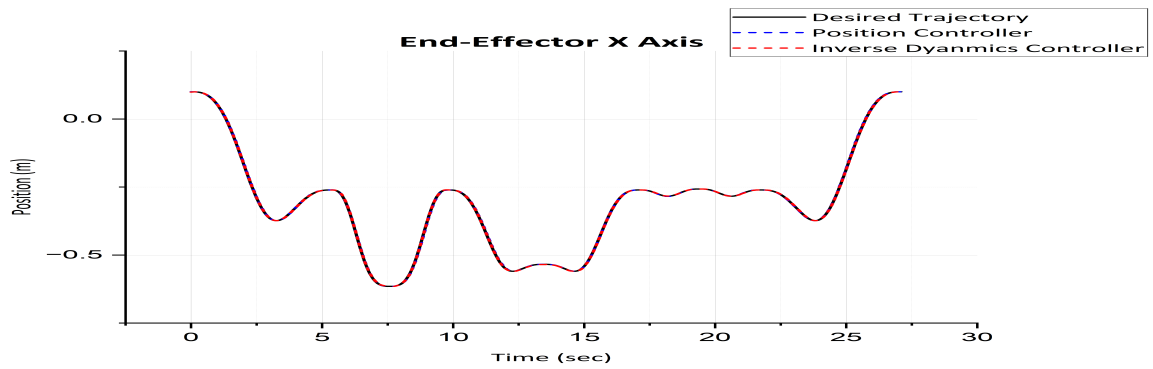


Figure 2.19: Result: end-effector x-axis / 1 rad/s / 1kg.

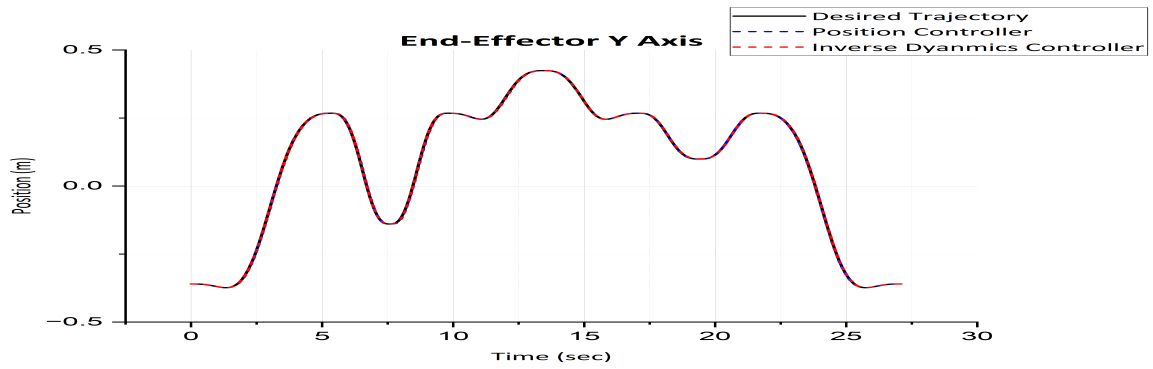


Figure 2.20: Result: end-effector y axis / 1 rad/s / 1kg.

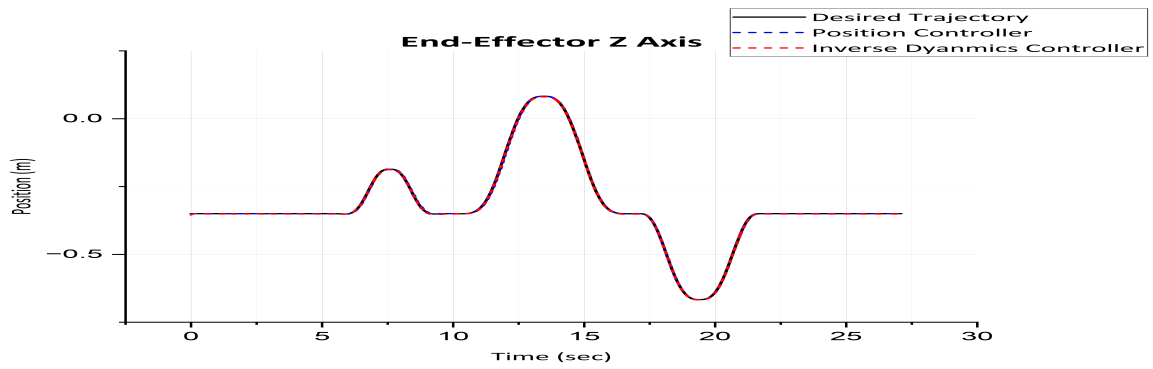


Figure 2.21: Result: end-effector z-axis / 1 rad/s / 1kg.

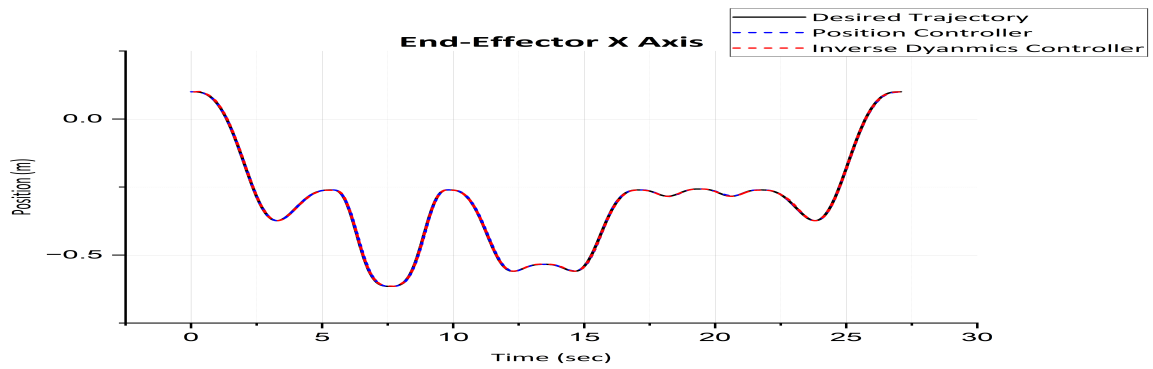


Figure 2.22: Result: end-effector x-axis / 1 rad/s / 2kg.

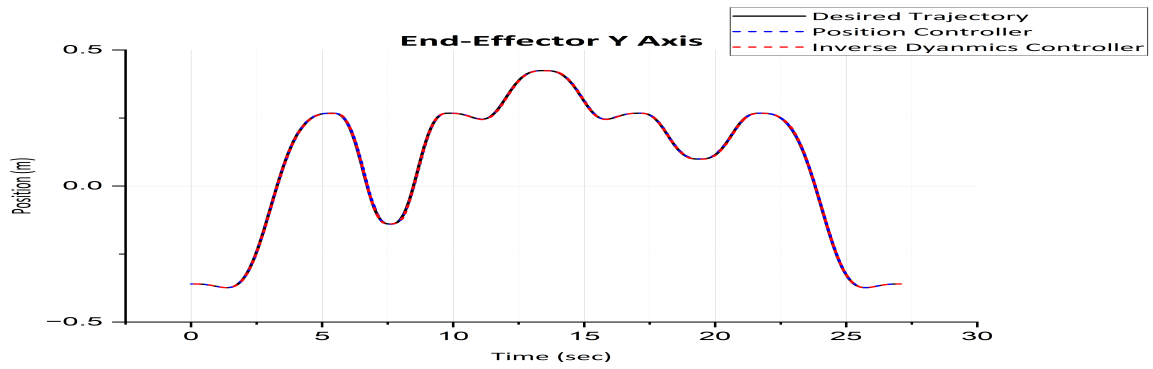


Figure 2.23: Result: end-effector y axis / 1 rad/s / 2kg.

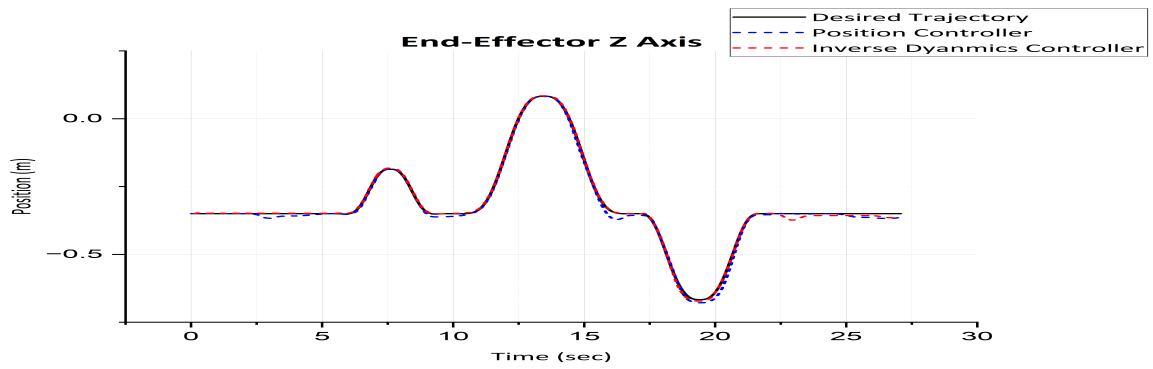


Figure 2.24: Result: end-effector z-axis / 1 rad/s / 2kg.

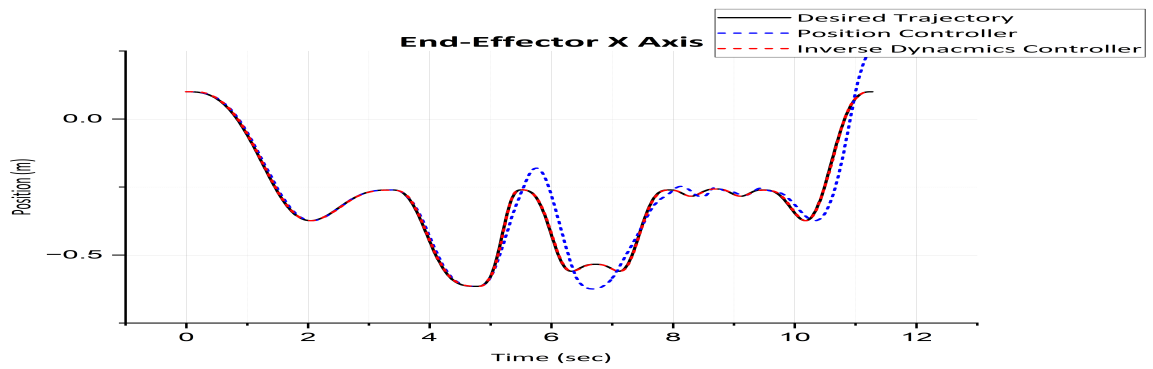


Figure 2.25: Result: end-effector x-axis / 3 rad/s / 0kg.

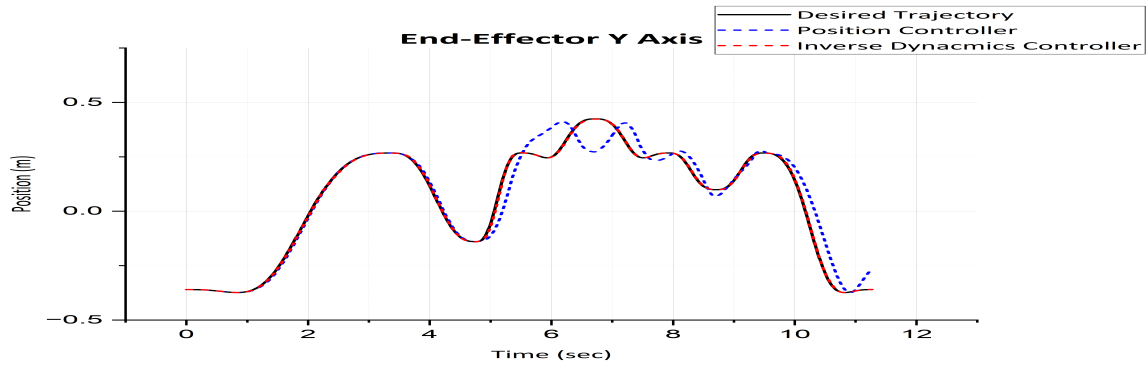


Figure 2.26: Result: end-effector y axis / 3 rad/s / 0kg.

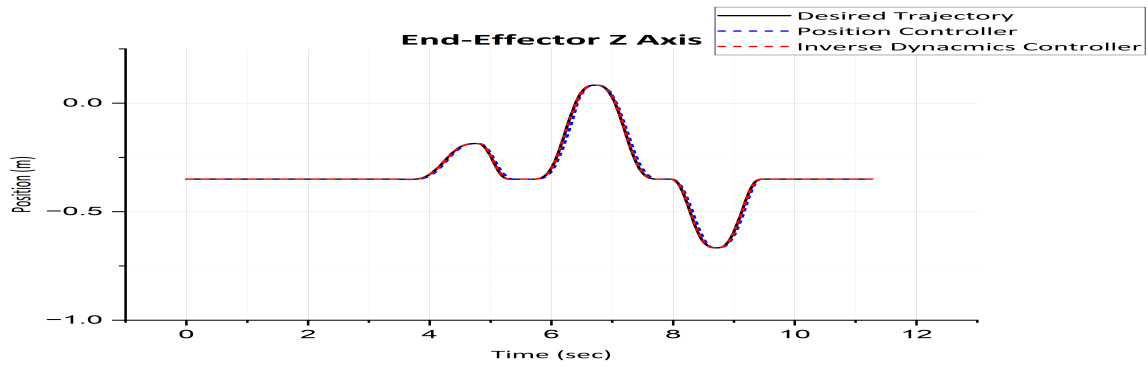


Figure 2.27: Result: end-effector z-axis / 3 rad/s / 0kg.

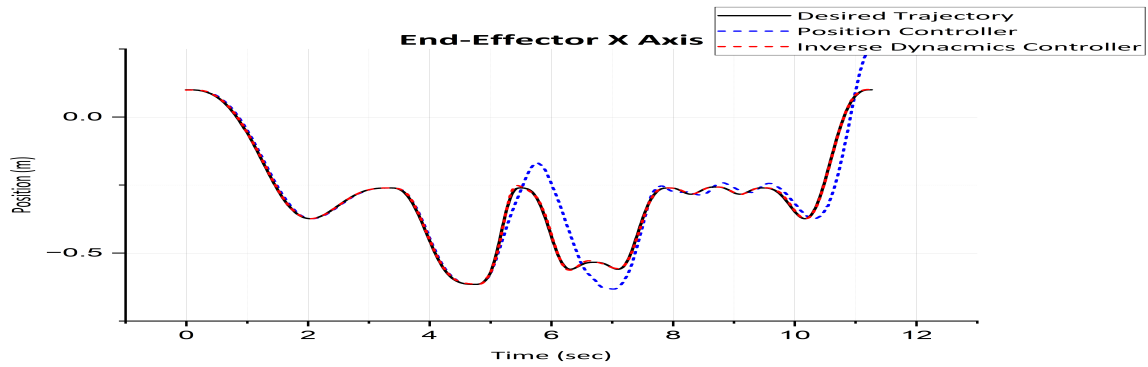


Figure 2.28: Result: end-effector x-axis / 3 rad/s / 1kg.

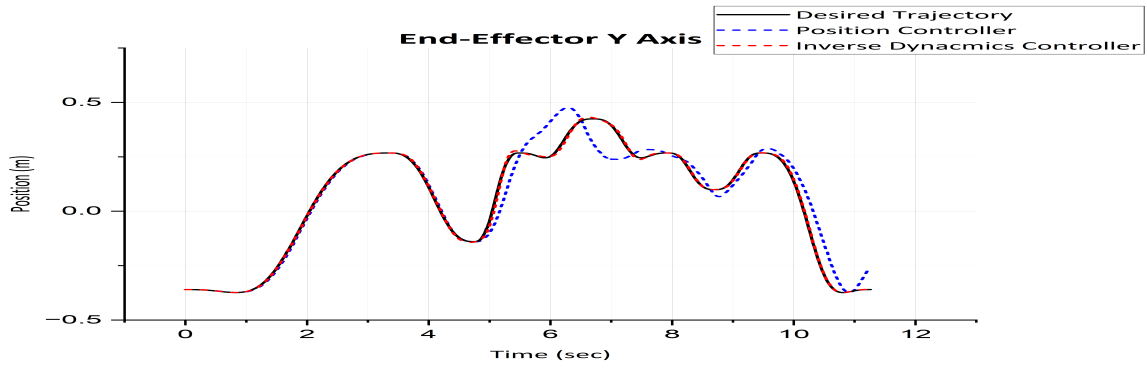


Figure 2.29: Result: end-effector y axis / 3 rad/s / 1kg.

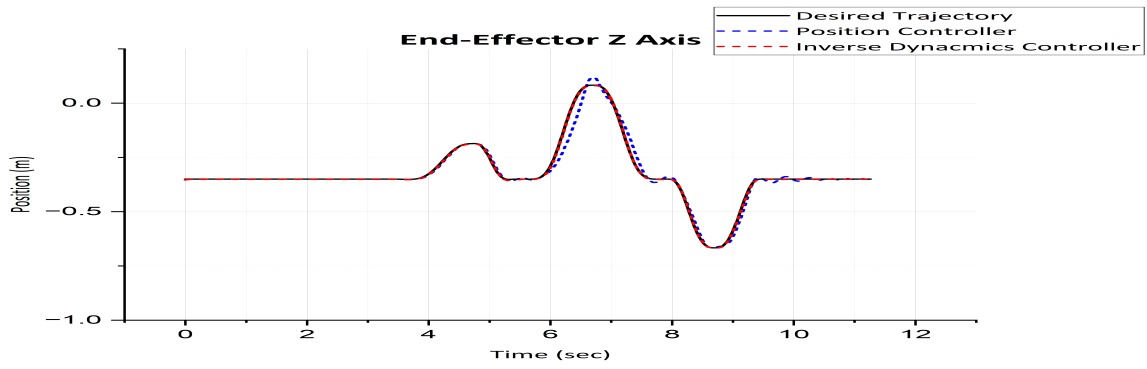


Figure 2.30: Result: end-effector z-axis / 3 rad/s / 1kg.

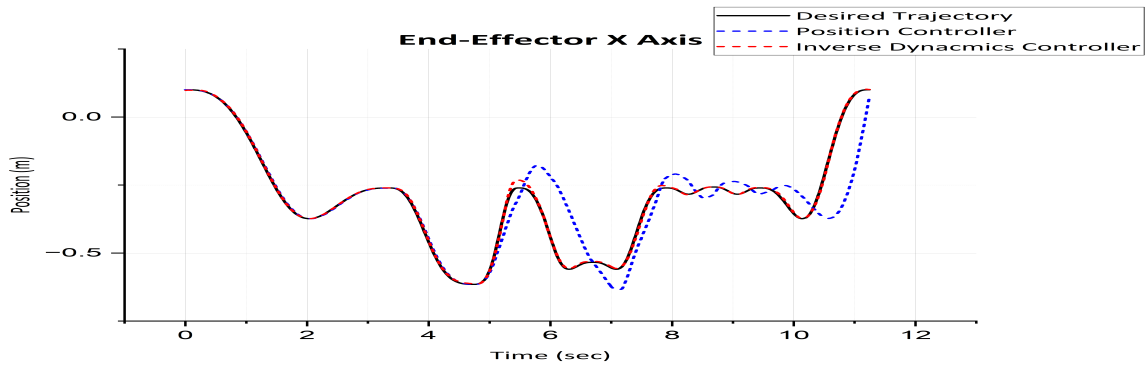


Figure 2.31: Result: end-effector x-axis / 3 rad/s / 2kg.

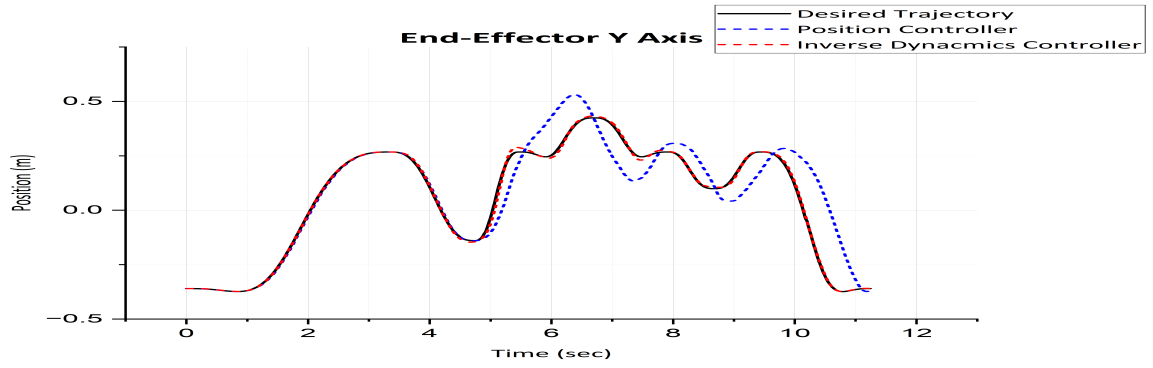


Figure 2.32: Result: end-effector y axis / 3 rad/s / 2kg.

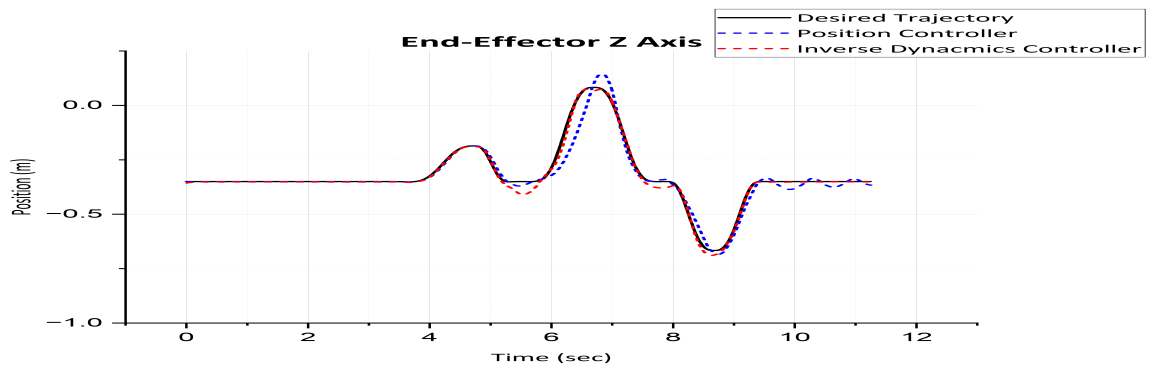


Figure 2.33: Result: end-effector z-axis / 3 rad/s / 2kg.

2.3.2 Impedance Controller Implementation

Pseudocode 3 Dual-Arm Impedance Control

Require: $\mathbf{M}_d, \mathbf{B}_d, \mathbf{K}_d, \mathbf{K}_{\text{sync}}, \mathbf{x}_d, \mathbf{x}, \dot{\mathbf{x}}, \mathbf{F}_{\text{ext}}$

```

1:  $\mathbf{F}_{\text{left}} \leftarrow \mathbf{0}, \mathbf{F}_{\text{right}} \leftarrow \mathbf{0}$  {Initialize forces}
2: for each arm (left, right) do
3:    $\mathbf{F}_d \leftarrow \mathbf{M}_d \ddot{\mathbf{x}}_d + \mathbf{B}_d (\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_d (\mathbf{x}_d - \mathbf{x})$  {Compute desired interaction force}
4: end for
5:  $\mathbf{F}_{\text{sync}} \leftarrow \mathbf{K}_{\text{sync}} (\mathbf{x}_{\text{left}} - \mathbf{x}_{\text{right}})$  {Synchronization force}
6:  $\mathbf{F}_{\text{left}} \leftarrow \mathbf{F}_d + \mathbf{F}_{\text{sync}}$ 
7:  $\mathbf{F}_{\text{right}} \leftarrow \mathbf{F}_d - \mathbf{F}_{\text{sync}}$ 
8: for each arm (left, right) do
9:    $\boldsymbol{\tau} \leftarrow \mathbf{J}^T \mathbf{F}$  {Compute joint torques using Jacobian transpose}
10: end for
11: return  $\boldsymbol{\tau}_{\text{left}}, \boldsymbol{\tau}_{\text{right}}$ 

```

The dual-arm impedance controller is implemented as outlined in Algorithm 3. This controller initializes the interaction forces for both the left and right arms and iteratively computes the desired interaction forces based on the specified mass (\mathbf{M}_d), damping (\mathbf{B}_d), and stiffness (\mathbf{K}_d) parameters. Additionally, a synchronization stiffness (\mathbf{K}_{sync}) is employed to ensure coordinated movements between the two arms. The desired interaction force \mathbf{F}_d is calculated to achieve the desired dynamic behavior, while the synchronization force \mathbf{F}_{sync} maintains balance and coordination between the arms by minimizing the positional difference between them. Subsequently, the joint torques for each arm are determined using the Jacobian transpose method, which translates the computed forces into the necessary torques for precise actuator control. This systematic approach enables the robotic system to perform smooth and efficient trajectory tracking while effectively responding to external disturbances, thereby enhancing overall system stability and performance.

2.3.3 Jerk-Optimized Dynamic Model Predictive Control

Unlike the typical formulation for manipulation, minimizing velocity and acceleration is inefficient for optimization when using proprioceptive actuators due to their fast joint speeds. Instead, minimizing torque and jerk proves to be more effective for these types of actuators. Additionally, by incorporating the current position and/or velocity as constraints, the controller demonstrates excellent recovery performance when deviating from the nominal trajectory.

In this study, we advance the dynamic MPC framework introduced in [LSB23] by incorporating jerk terms to facilitate smooth and efficient motion planning while explicitly accounting for system dynamics. The proposed formulation incorporates jerk minimization within the cost function and imposes jerk constraints to ensure precise trajectory tracking and seamless transitions, particularly during high-speed motions. Moreover, the MPC framework leverages the current position as an initial constraint, enabling rapid, safe, and smooth operation, as well as robust trajectory recovery under dynamic conditions. This methodology is specifically designed to meet the unique requirements of proprioceptive manipulators.

The objective of Jerk-Optimized Dynamic MPC is to minimize a cost function that combines task tracking error, control effort, and jerk (rate of change of acceleration):

$$J = \sum_{k=i}^{i+N_p} \left[(\hat{\theta}_k - g(q_k))^T Q_e (\hat{\theta}_k - g(q_k)) + u_k^T Q_u u_k + j_k^T Q_j j_k \right], \quad (2.34)$$

where $\hat{\theta}_k$: Desired task-space state at time step k , $g(q_k)$: Actual task-space state computed from forward kinematics, u_k : Torque input at time step k , $j_k = (\ddot{q}_k - \ddot{q}_{k-1})/\Delta t$: Jerk, Q_e, Q_u, Q_j : Weighting matrices for task error, torque, and jerk, respectively.

To address the nonlinearity of $g(q_k)$, we apply a first-order Taylor expansion to linearize the function as follows:

$$\hat{\theta}_k - g(q_k) \approx \hat{J}_k (\hat{q}_k - q_k), \quad (2.35)$$

where $J_k = \left. \frac{\partial g}{\partial q} \right|_{\hat{q}_k}$ denotes the Jacobian matrix evaluated at the nominal configuration \hat{q}_k .

Substituting equation 2.35 into equation 2.34, the tracking error term becomes:

$$(\hat{\theta}_k - g(q_k))^T Q_e (\hat{\theta}_k - g(q_k)) \approx q_k^T \hat{J}_k^T Q_e \hat{J}_k q_k - 2\dot{q}_k^T \hat{J}_k^T Q_e \hat{J}_k q_k + \text{constant}. \quad (2.36)$$

This reformulation allows the cost function to be expressed in a quadratic form, making it suitable for QP solvers.

The resulting QP optimization problem is:

$$\begin{aligned} \min_{z_k} \quad & z_k^T W z_k + 2w^T z_k, \\ \text{subject to} \quad & x_{k+1} = A_k x_k + B_k u_k + r_k, \\ & u_{\min} \leq u_k \leq u_{\max}, \\ & q_{\min} \leq q_k \leq q_{\max}, \\ & \dot{q}_{\min} \leq \dot{q}_k \leq \dot{q}_{\max}, \\ & \ddot{q}_{\min} \leq \ddot{q}_k \leq \ddot{q}_{\max}, \\ & q_0 = q_{\text{current}}, \\ & \dot{q}_0 = \dot{q}_{\text{current}}. \end{aligned} \quad (2.37)$$

Here, $z_k = [x_k^T, u_k^T]^T$ is the decision variable, and W and w are the quadratic and linear coefficients derived from the cost function and constraints.

This formulation effectively integrates jerk minimization, torque control, and initial state constraints, thereby ensuring efficient recovery and smooth trajectory tracking for manipulators equipped with proprioceptive actuators. As demonstrated in Figure 2.34, Model Predictive Control (MPC) accurately tracks the quintic spline reference trajectory while generating a smoother torque trajectory, as illustrated in Figure 2.35.

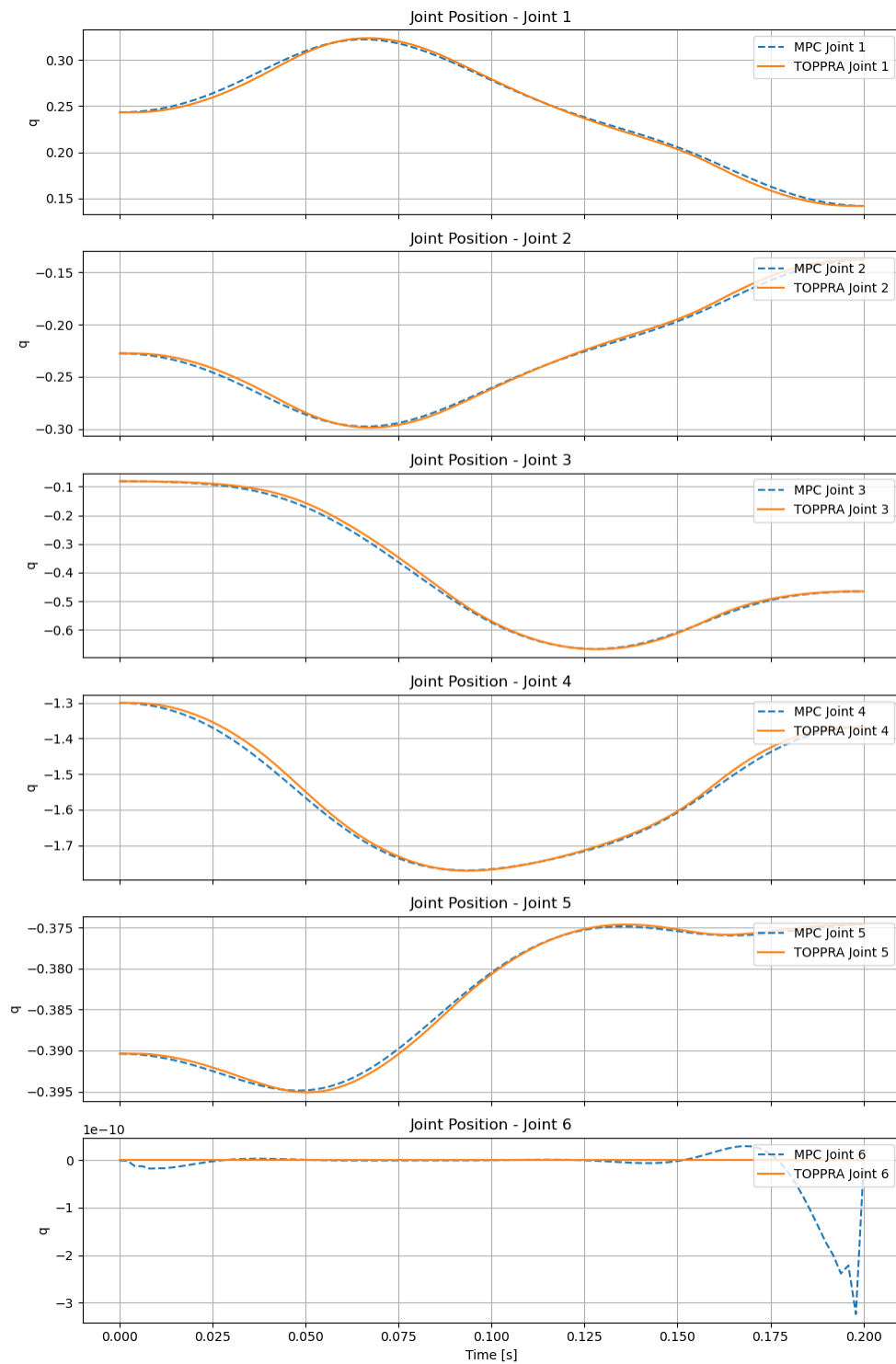


Figure 2.34: Comparison of joint positions between the reference trajectory and the MPC-generated trajectory.

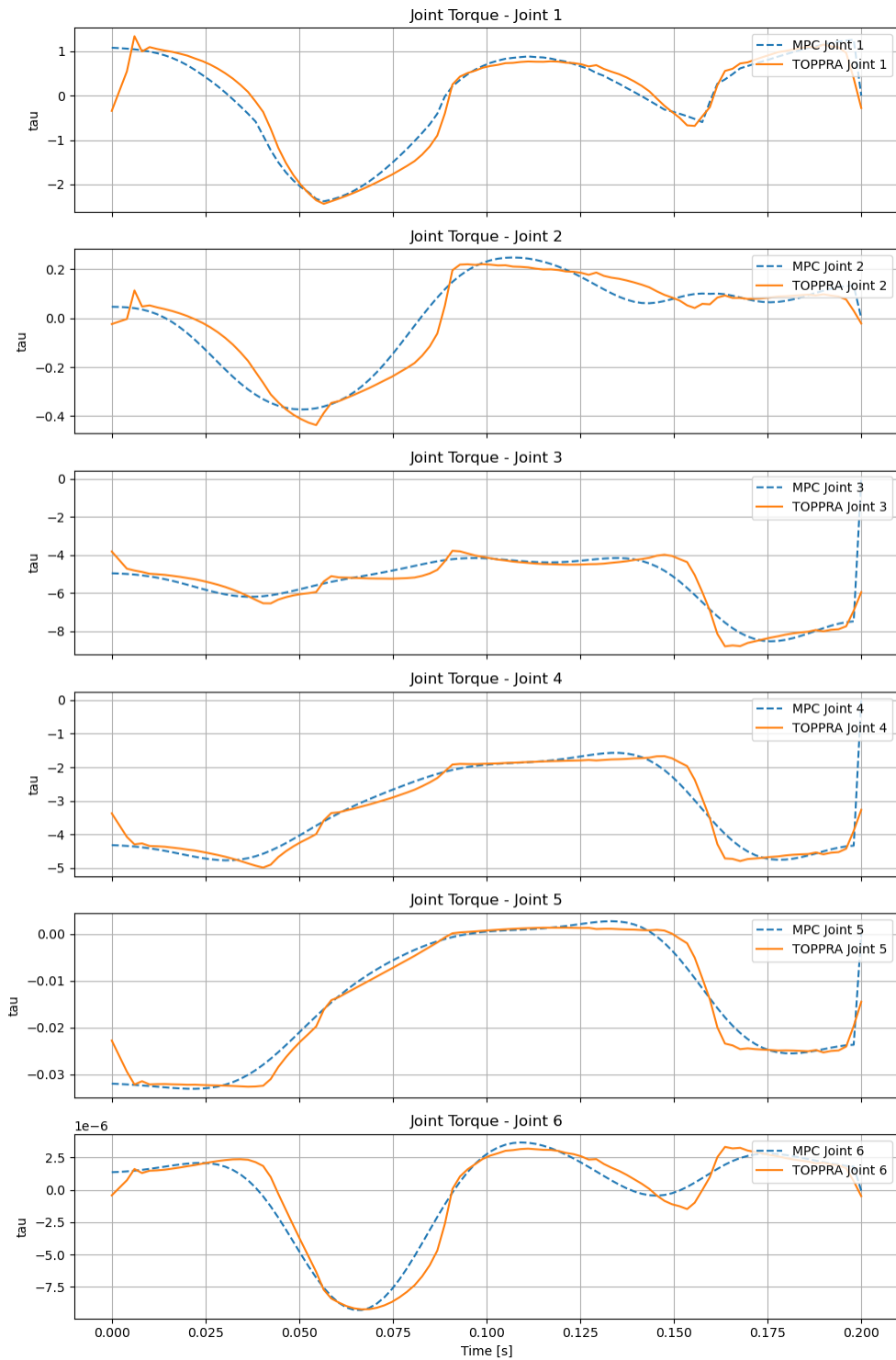


Figure 2.35: Comparison of joint torques between the reference trajectory and the MPC-generated trajectory.

2.4 Motion Planning: Time-Optimal and Jerk-Minimized Trajectory Planning

When planning trajectories for a manipulator equipped with high-speed proprioceptive actuators, the primary focus often shifts from strict time optimality to generating smoother trajectories. This emphasis on smoothness resonates with well-documented principles of human motor control: the human central nervous system naturally tends towards maximizing smoothness in limb movements. In their seminal work, Flash and Hogan [FH85b] observed human arm motions and demonstrated that humans generate smooth point-to-point movements by minimizing jerk, defined as the rate of change of acceleration. Their model captured this behavior by formulating an optimization problem that minimized the L_2 -norm of the jerk along the trajectory.

Subsequent research by Yazdani [YGH12] suggested that minimizing the L_∞ -norm of the jerk produces a bang-bang type control policy, more closely aligning with specific biological control strategies employed by the central nervous system. Drawing inspiration from these studies, we implemented a time-optimal and jerk-minimized trajectory planning framework. This approach aims to produce trajectories that are both dynamically feasible and inherently smooth, ensuring that each joint follows a path that balances speed with ease of execution.

2.4.1 Trajectory Optimization Formulation

We consider a robotic manipulator with n_j joints. The motion is discretized into N time steps, and let $q[i] \in \mathbb{R}^{n_j}$ represent the joint configuration at the i -th time step, with $i \in \{0, \dots, N - 1\}$. The total execution time is $T > 0$, and the time increment is defined as $\Delta t = T/(N - 1)$.

- **Decision variables:**

- All decision variables are collected into a single vector:

$$x = \begin{bmatrix} T \\ q[0, 1] \\ q[0, 2] \\ \vdots \\ q[0, n_j] \\ q[1, 1] \\ \vdots \\ q[N - 1, n_j] \end{bmatrix} \in \mathbb{R}^{1+Nn_j}. \quad (2.38)$$

By choosing T and $q[i, j]$ for each time step i and joint j , the entire trajectory is fully determined.

- **Velocity and acceleration approximations:**

- Velocities and accelerations are approximated via finite differences. For $1 \leq i \leq N - 2$ and each joint j :

$$\dot{q}[i, j] \approx \frac{q[i + 1, j] - q[i - 1, j]}{2\Delta t}, \quad \ddot{q}[i, j] \approx \frac{q[i + 1, j] - 2q[i, j] + q[i - 1, j]}{(\Delta t)^2}. \quad (2.39)$$

- At the boundaries ($i = 0$ and $i = N - 1$), one-sided differences are employed.

- **Cost function:**

- The cost consists of two terms: one to minimize jerk and thus ensure smoothness, and another to minimize the total execution time T .
- Jerk is the time derivative of acceleration. Discretely, for $0 \leq i \leq N - 3$:

$$\text{Jerk}[i, j] \approx \frac{\ddot{q}[i + 1, j] - \ddot{q}[i, j]}{\Delta t}. \quad (2.40)$$

- Define the jerk cost term J_{jerk} as:

$$J_{\text{jerk}} = \sum_{i=0}^{N-3} \sum_{j=1}^{n_j} \left(\frac{\ddot{q}[i + 1, j] - \ddot{q}[i, j]}{\Delta t} \right)^2. \quad (2.41)$$

– The complete objective function is:

$$\text{Cost} = \lambda_{\text{jerk}} J_{\text{jerk}} + \lambda_{\text{time}} T, \quad (2.42)$$

where $\lambda_{\text{jerk}} \geq 0$ and $\lambda_{\text{time}} \geq 0$ control the trade-off between smoothness and speed.

• **Constraints:**

– *Boundary conditions:*

$$q[0] = q_{\text{start}}, \quad q[N-1] = q_{\text{end}}, \quad (2.43)$$

and zero initial/final velocities and accelerations:

$$\dot{q}[0] = 0, \quad \dot{q}[N-1] = 0, \quad \ddot{q}[0] = 0, \quad \ddot{q}[N-1] = 0. \quad (2.44)$$

– *Joint limits:*

$$q_{\min,j} \leq q_{i,j} \leq q_{\max,j}, \quad \forall i, j. \quad (2.45)$$

– *Torque constraints:* Let $\tau[i] \in \mathbb{R}^{n_j}$ be the torque at time step i . Using the Recursive Newton-Euler Algorithm (RNEA):

$$\tau[i] = M(q[i])\ddot{q}[i] + C(q[i], \dot{q}[i])\dot{q}[i] + G(q[i]), \quad (2.46)$$

where M is the inertia matrix, C the Coriolis/centrifugal matrix, and G the gravity vector. The torque limits are:

$$\tau_{\min,j} \leq \tau[i, j] \leq \tau_{\max,j}, \quad \forall i, j. \quad (2.47)$$

Final Nonlinear Program:

Combining all conditions, we obtain a nonlinear program (NLP):

$$\min_x \quad \lambda_{\text{jerk}} J_{\text{jerk}} + \lambda_{\text{time}} T \quad (2.48)$$

subject to:

$$q[0] = q_{\text{start}}, \quad q[N - 1] = q_{\text{end}}, \quad (2.49)$$

$$\dot{q}[0] = 0, \quad \dot{q}[N - 1] = 0, \quad \ddot{q}[0] = 0, \quad \ddot{q}[N - 1] = 0, \quad (2.50)$$

$$q_{\min,j} \leq q_{i,j} \leq q_{\max,j}, \quad \forall i, j, \quad (2.51)$$

$$\tau_{\min,j} \leq \tau[i, j] \leq \tau_{\max,j}, \quad \forall i, j. \quad (2.52)$$

This NLP can be solved using state-of-the-art nonlinear optimization solvers (e.g., IPOPT) within a fraction of a second, especially when using a predefined reference path that simplifies the search space. In our experiments, the presence of a known reference path and efficient warm-start strategies enabled solution times to remain under one second. Moreover, traditional nonlinear optimization frameworks based on Sequential Quadratic Programming (SQP) or Sequential Linear Programming (SLP) can also be employed to converge to a feasible, near-optimal solution rapidly.

2.4.2 Experimental Results

To evaluate the performance and characteristics of the proposed trajectory optimization method, we consider a scenario with three key points in Cartesian space: a starting point, a via point, and an ending point. Each experimental configuration highlights different aspects of trajectory generation under various constraints.

First, we generate a SE(3) quintic spline path in Cartesian space and derive the corresponding joint-space path through inverse kinematics. This baseline representation, shown in Figure 2.36, establishes a smooth geometric reference that will serve as a foundation for subsequent trajectory parameterizations and optimizations.

Next, we employ TOPP-RA to parameterize the reference joint-space path, imposing velocity and acceleration limits to ensure feasibility. Figure 2.37 illustrates how the given path can be time-parameterized to respect kinematic constraints using TOPP-RA, producing

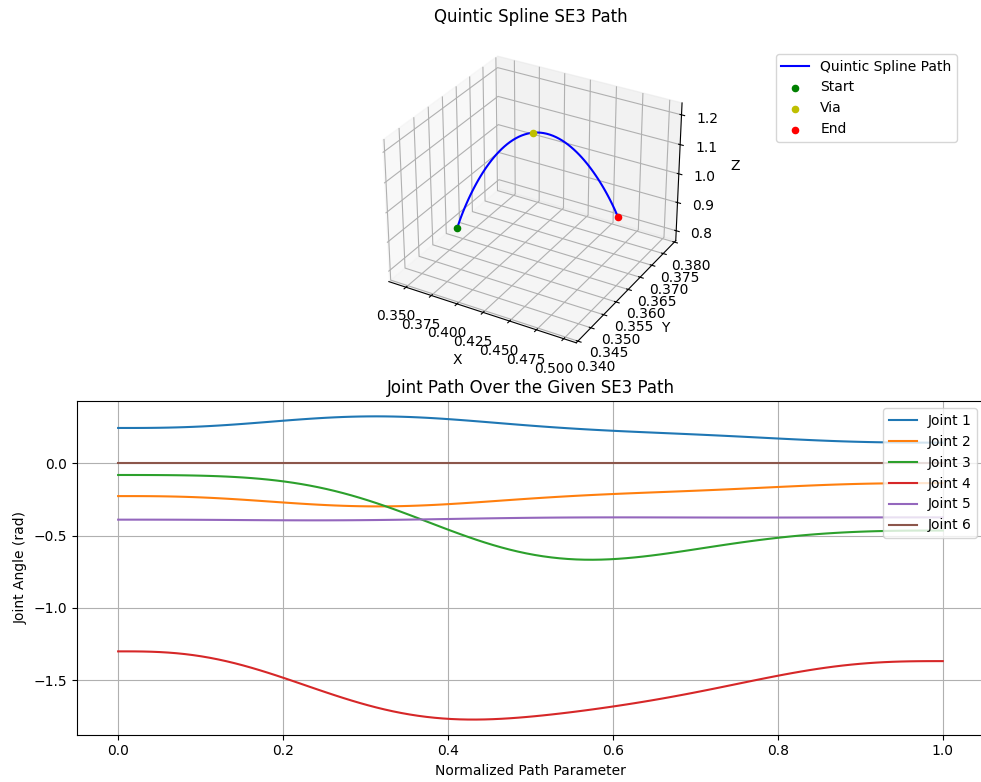


Figure 2.36: SE(3) quintic spline path and corresponding joint path derived via inverse kinematics.

a dynamically realizable trajectory.

Building on this, we apply our time-optimal and jerk-minimized trajectory optimization formulation directly to the reference path. As depicted in Figure 2.38, the resulting trajectory achieves a balance between rapid execution and minimized jerk, demonstrating smooth yet efficient motion suitable for high-speed robotic tasks.

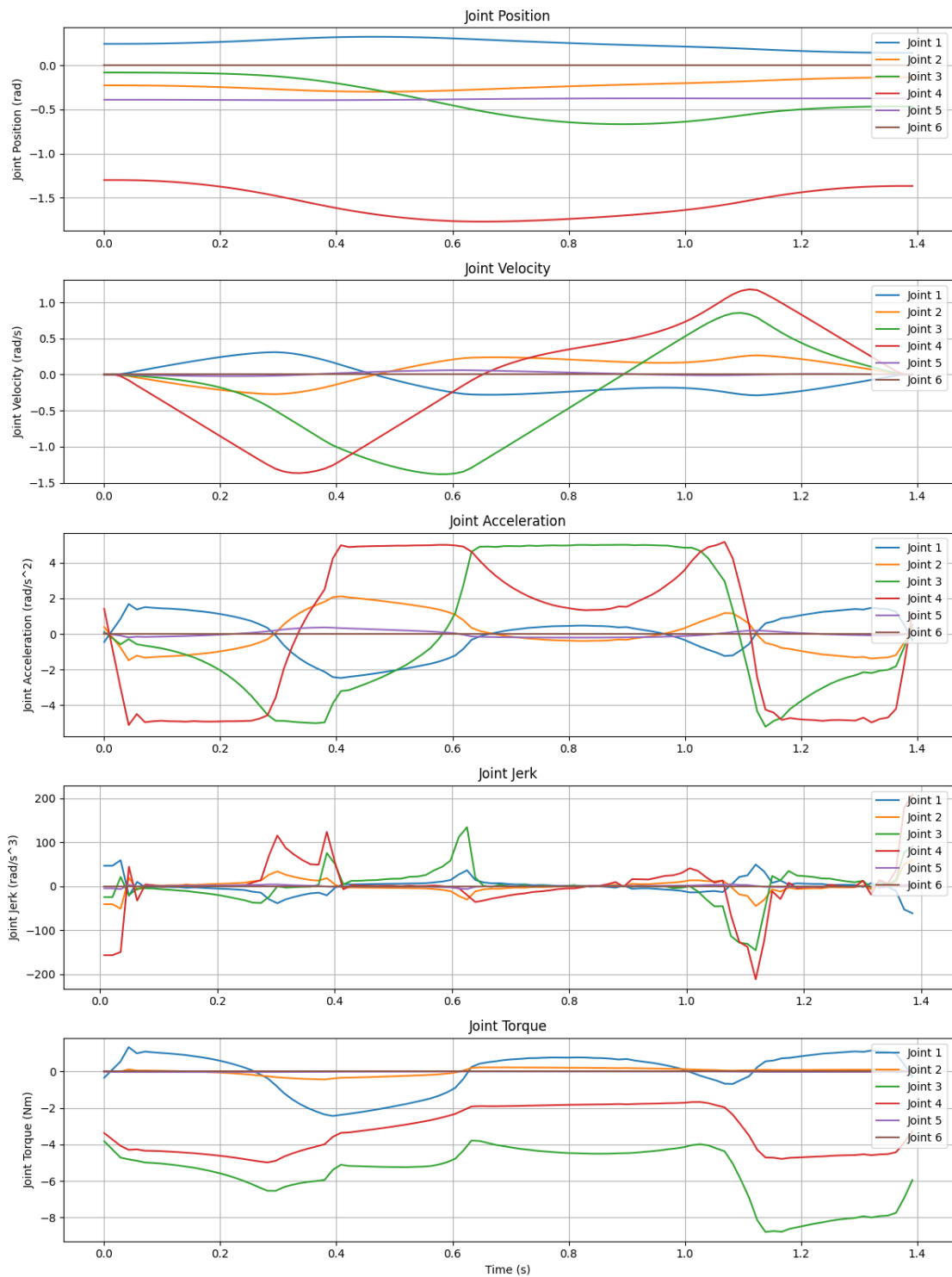


Figure 2.37: Time-parameterization of the reference joint path using TOPP-RA, respecting velocity and acceleration limits.

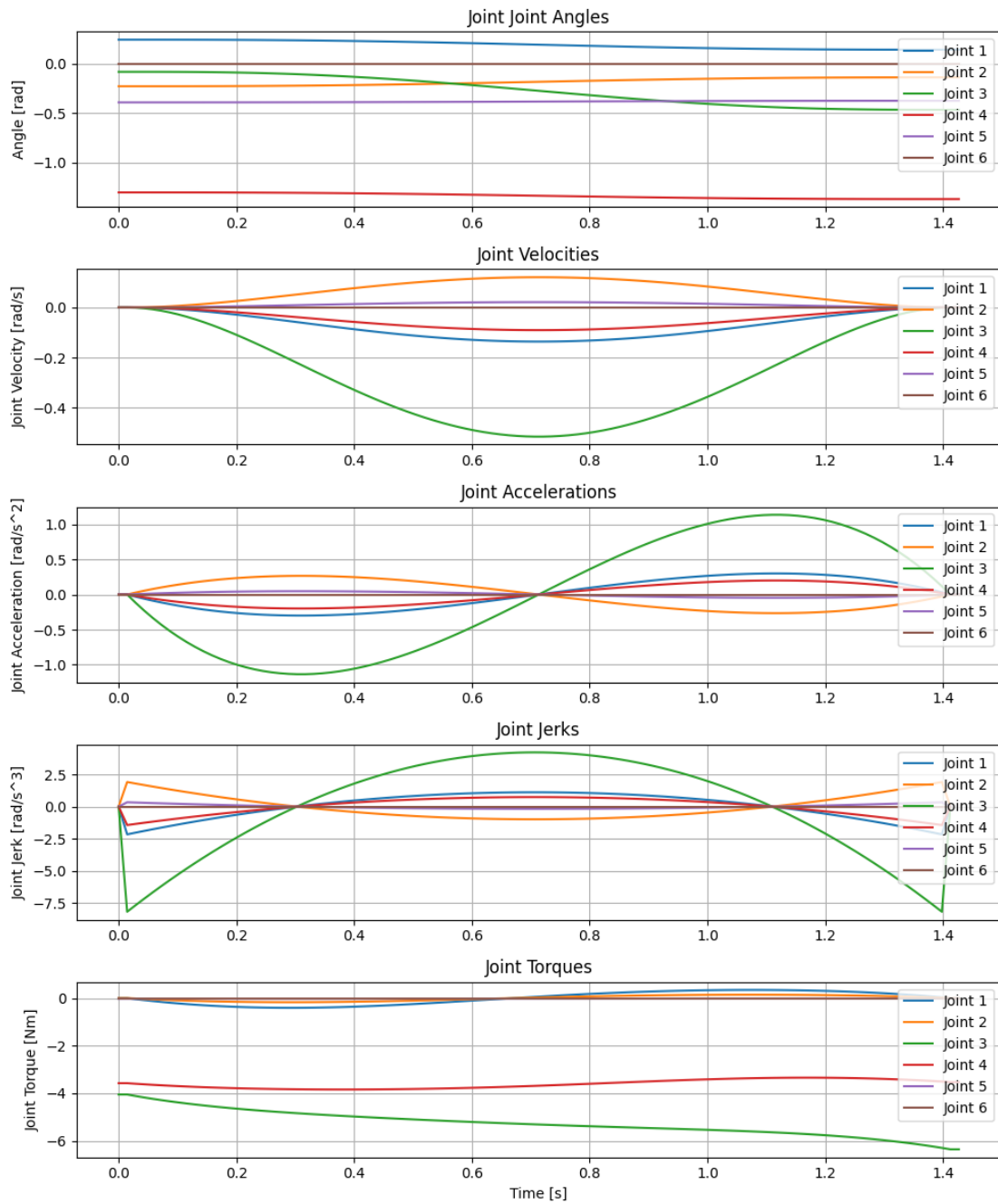


Figure 2.38: Time-optimal and jerk-minimized trajectory derived from the reference path, balancing execution time and smoothness.

We further compare two approaches for minimum jerk trajectory generation. The first approach, shown in Figure 2.39, applies a minimum jerk formulation twice and enforces that the via point is reached with zero velocity. This produces a trajectory with a distinct stopping behavior at the intermediate waypoint. In contrast, Figure 2.40 presents a minimum jerk trajectory derived through an optimization process that does not enforce zero velocity at the via point, resulting in a continuously smooth motion profile that passes through the intermediate waypoint without halting.

Finally, to assess the dynamic feasibility and overall smoothness of these trajectories, we examine and compare their resulting torques and jerks. Figure 2.41 and Figure 2.42 respectively show the torque and jerk profiles obtained from the various trajectories, enabling a direct comparison of their dynamic characteristics. To provide a more quantitative summary, Figure 2.43 and Figure 2.44 present bar charts summarizing the final torque and jerk values for each tested trajectory, clearly illustrating the benefits of our optimization strategy in achieving low jerk and dynamically feasible torques.

By minimizing jerk and execution time while respecting kinematic and dynamic constraints, such a trajectory optimization framework is particularly valuable for dexterous, high-speed tasks. It ensures that resulting motions are both dynamically realizable and inherently smooth. By integrating insights from human motor control research and applying them to robotic manipulators, we achieve trajectories that not only move quickly but also emulate the natural, smooth qualities of human limb movements.

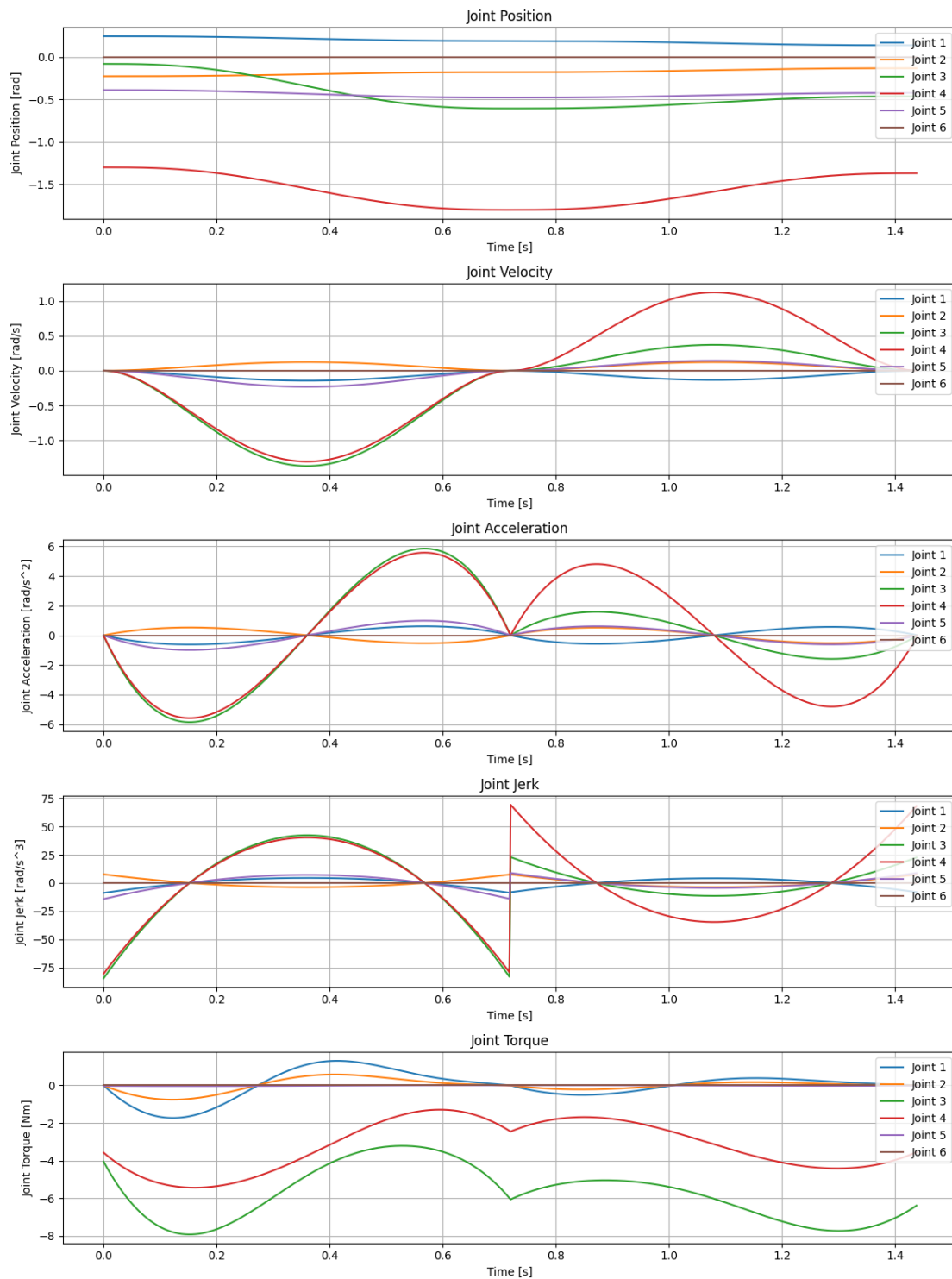


Figure 2.39: Minimum jerk trajectory obtained by applying the minimum jerk formulation twice, enforcing zero velocity at the via point.

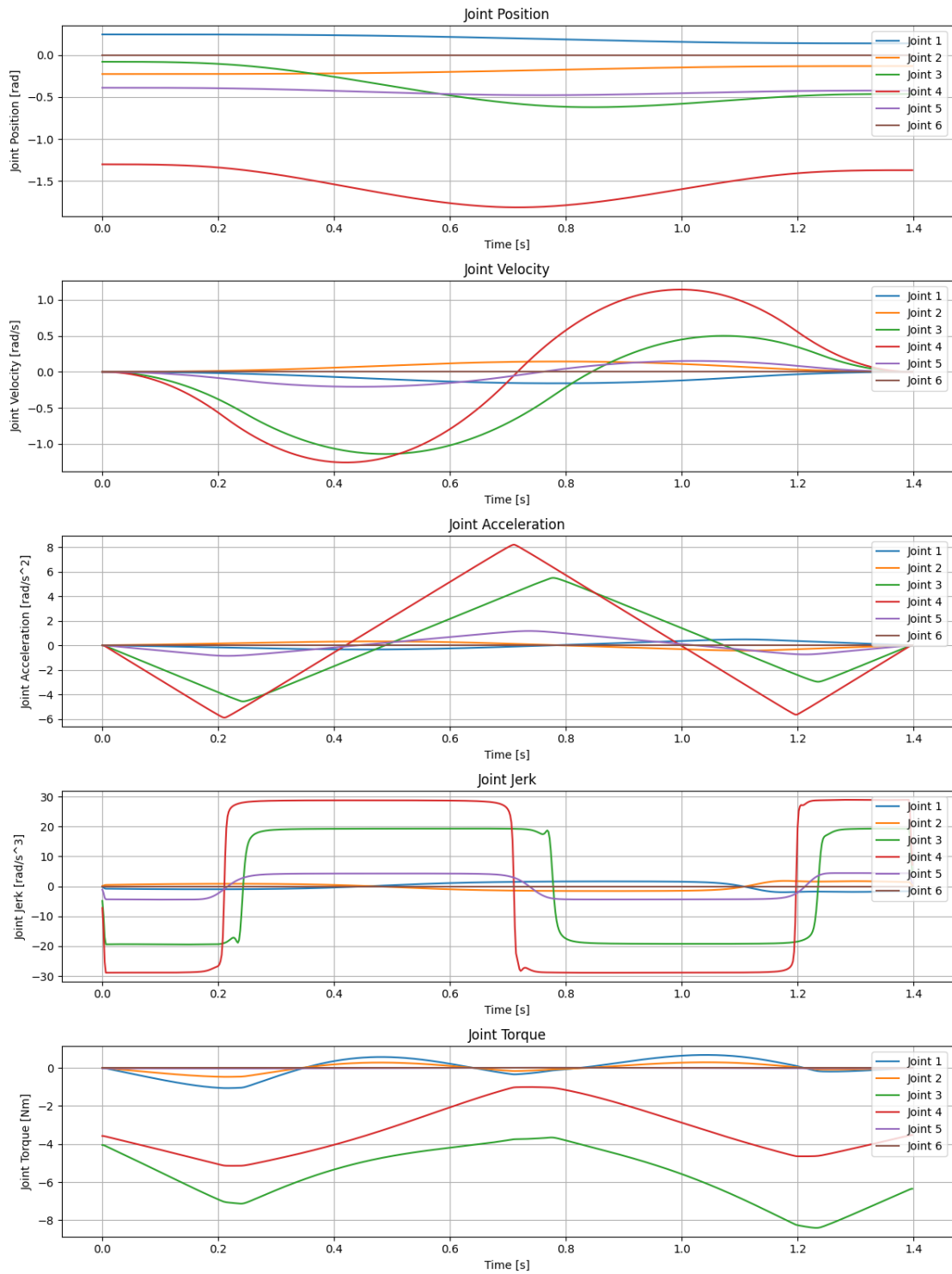


Figure 2.40: Optimized minimum jerk trajectory that passes through the via point without enforcing zero velocity, resulting in a continuously smooth profile.

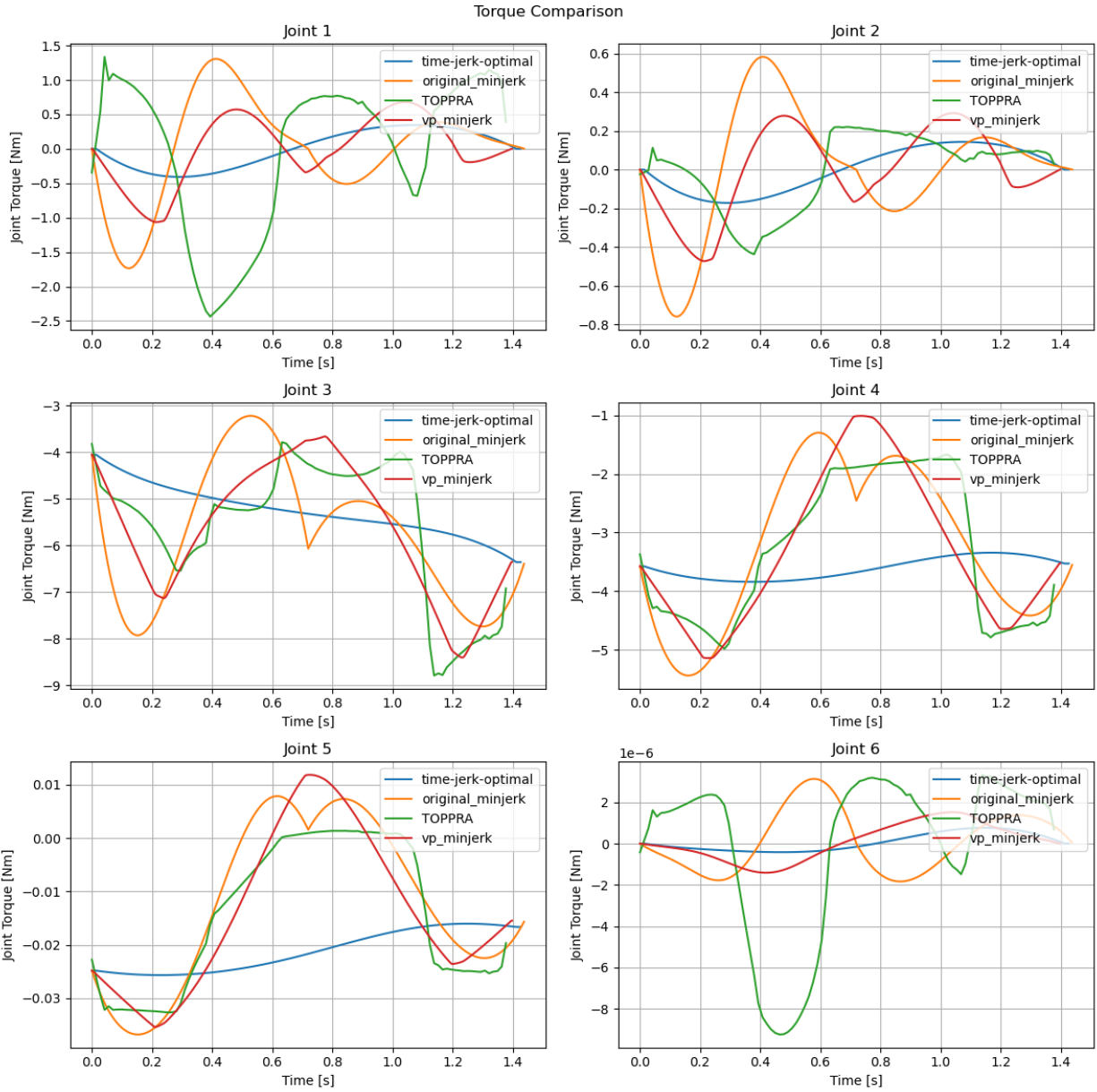


Figure 2.41: Final torque profiles obtained from the various tested trajectories, illustrating their dynamic feasibility.

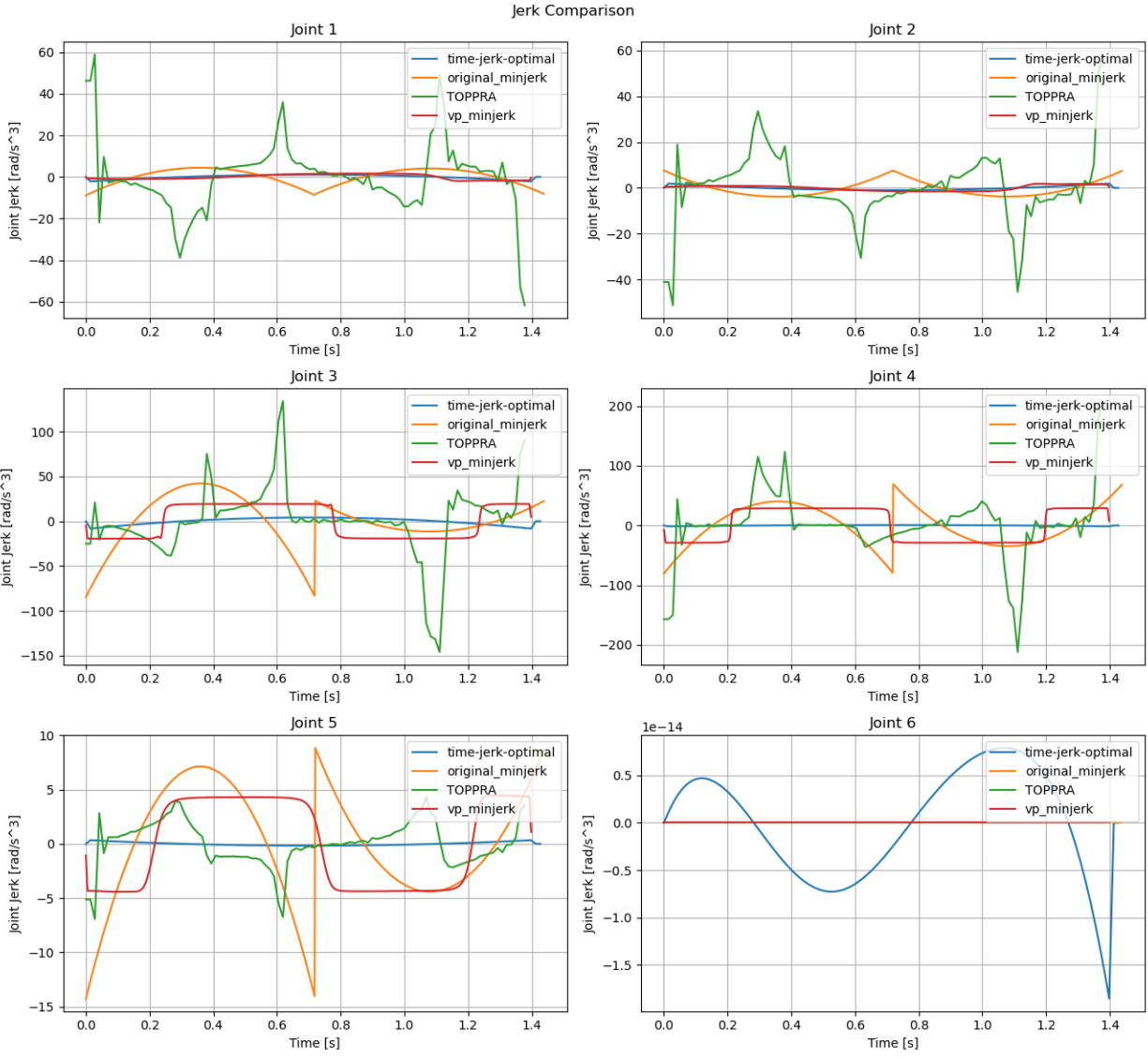


Figure 2.42: Final jerk profiles for the tested trajectories, demonstrating differences in motion smoothness.

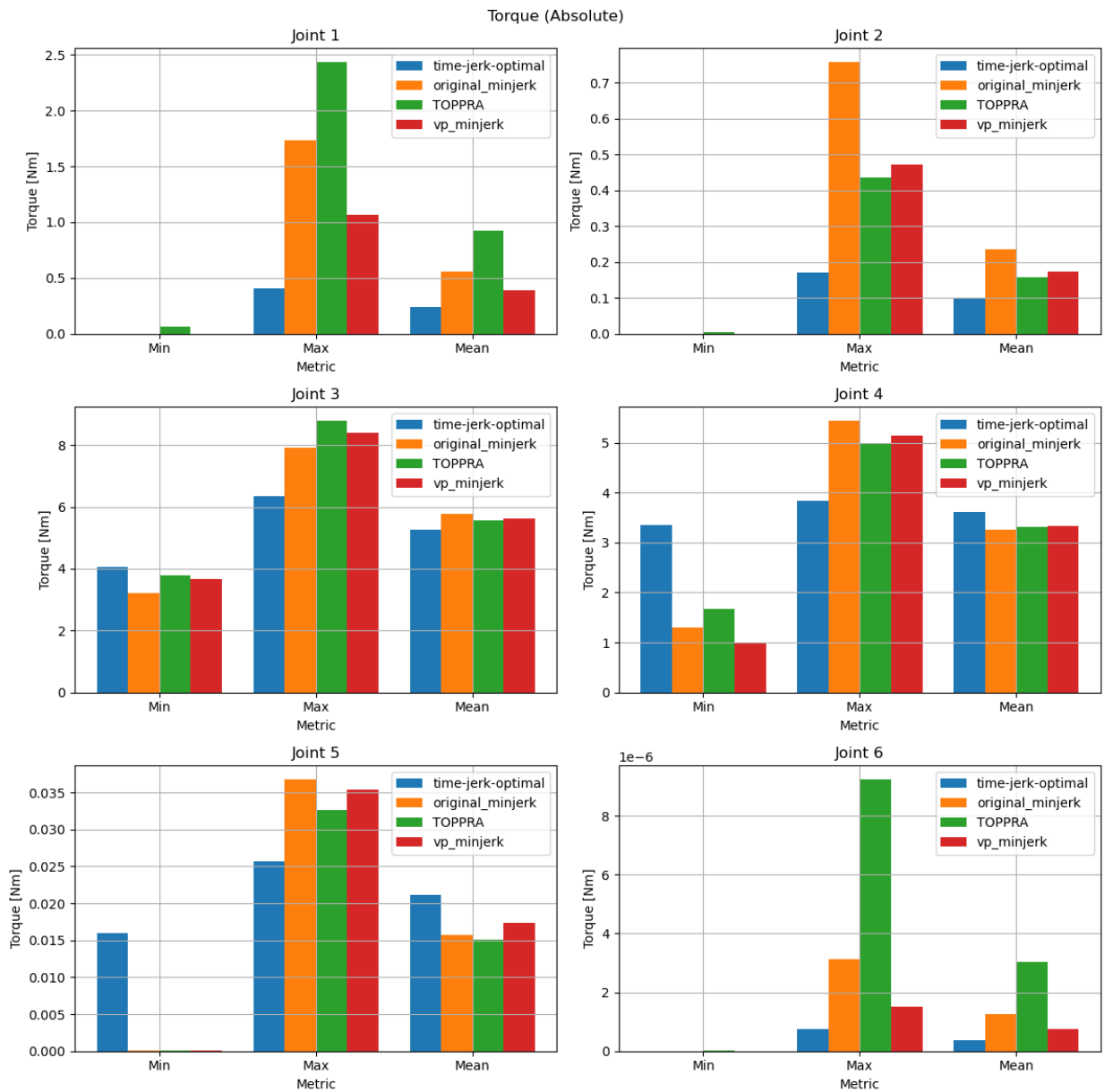


Figure 2.43: Bar chart summarizing the final torque values across all trajectories, enabling quantitative comparison.

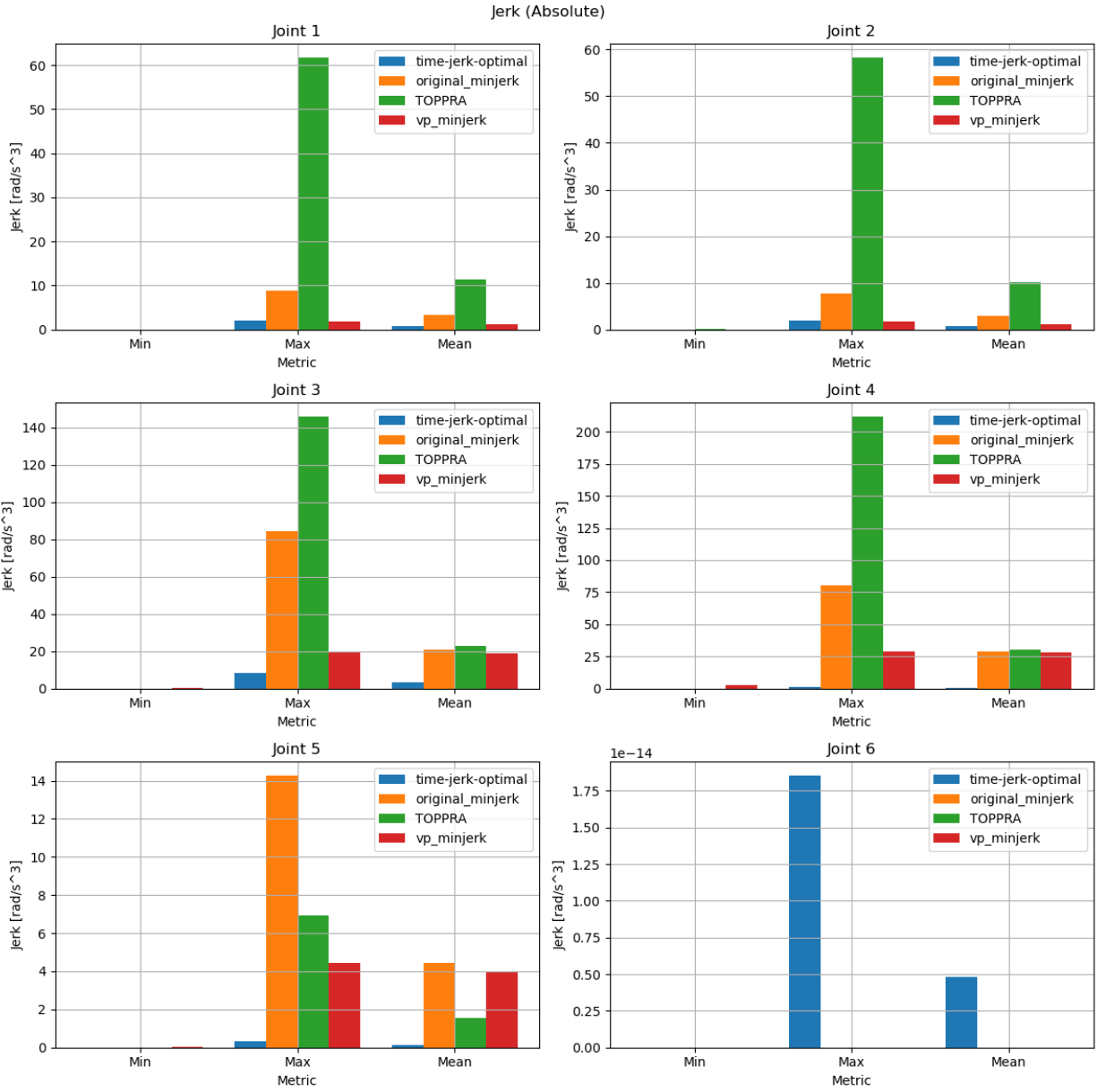


Figure 2.44: Bar chart summarizing the final jerk values for all trajectories, highlighting the benefits of our optimization strategy in minimizing jerk.

2.5 Summary

This chapter presented the design of a lightweight, low-inertia manipulator optimized to maximize the performance of proprioceptive actuators. Alongside the hardware development, appropriate control strategies and motion planning methodologies were proposed. For control, combining inverse dynamics and impedance controllers was identified as a robust solution for a wide range of applications, with a practical example introduced in the subsequent chapter. Additionally, the Jerk-Optimized Dynamic MPC approach was proposed. By minimizing tracking error alongside jerk and torque, the method enables the generation of fast yet smooth and safe trajectories.

For motion planning, while time-optimal trajectory generation is a common trend in domains utilizing conventional manipulators, this manipulator does not require explicit time-optimal considerations to achieve sufficiently high speeds. Instead, by adopting a formulation that minimizes jerk and incorporating dynamic constraints, the proposed approach efficiently generates trajectories with improved smoothness and feasibility.

Comparative analyses demonstrated the effectiveness of the proposed methods in producing trajectories that are not only smooth but also compliant with physical and dynamic constraints. The resulting trajectories exhibited superior adaptability and feasibility compared to the other methods. These results underscore the practical applicability of the proposed motion planning and control strategies in tasks demanding precision and safety, such as dexterous manipulation and human-robot collaboration.

The presented design, control methods, and motion planning techniques collectively establish a strong foundation for the development of advanced robotic manipulators capable of generating smooth, efficient, and dynamically consistent motions. This work significantly expands the capabilities of robotic manipulators in addressing complex and dynamic environments.

CHAPTER 3

YORI: Autonomous Cooking System Utilizing a Modular Robotic Kitchen and a Dual-Arm Proprioceptive Manipulator

3.1 System Design

In this chapter, we introduce our autonomous robotic cooking system, YORI (Yummy Operations Robot Initiative), which is designed to address the myriad challenges inherent in adapting traditional, human-centered cooking methods for autonomous robots. These challenges include developing kitchen tools and appliances suited for robotic manipulation, configuring the environment to maximize the robot’s operational reach, supporting a diverse and scalable menu, and ensuring stable, fully autonomous operation. An overview of the system is provided in Figure 3.1.

We leverage the intrinsic capabilities of robotics rather than relying on human-centric techniques, enabling more efficient culinary tasks. To achieve this, we developed specialized tools and appliances, configured a kitchen layout optimized for a dual-arm manipulator, and adopted a modular design that supports both general-purpose and specialized setups—enhancing menu variety and scalability. By integrating a dual-arm manipulator with proprioceptive actuators, we deliver a system that is immediately stable and functional, while also laying the foundation for future advancements in autonomy, stability, and versatility.

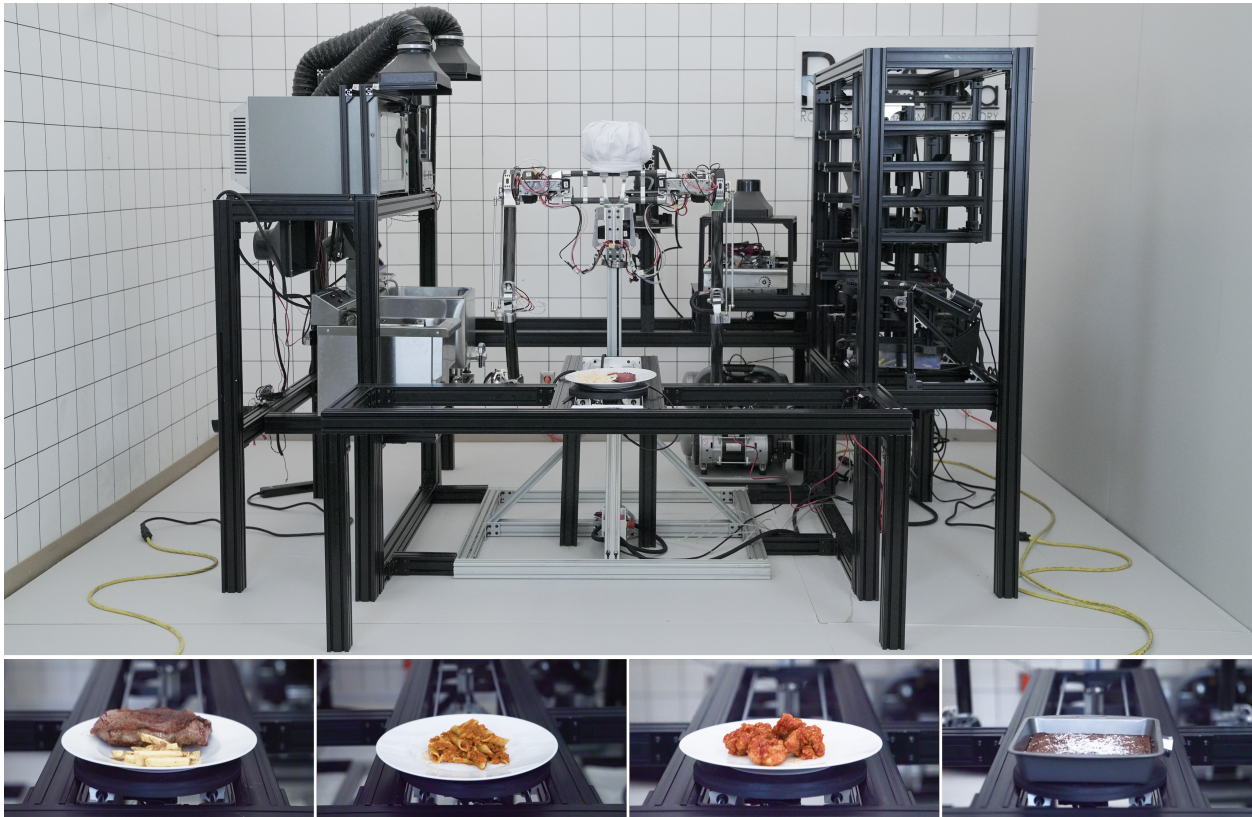


Figure 3.1: Overview of the YORI System: A centrally located dual-arm manipulator with proprioceptive actuators is surrounded by four modular kitchen units equipped with either customized or newly developed cooking tools and appliances. Shown below, from left to right, are dishes prepared by the system, including Steak Frites, Tomato Penne Pasta, Spicy Fried Chicken, and Brownie.

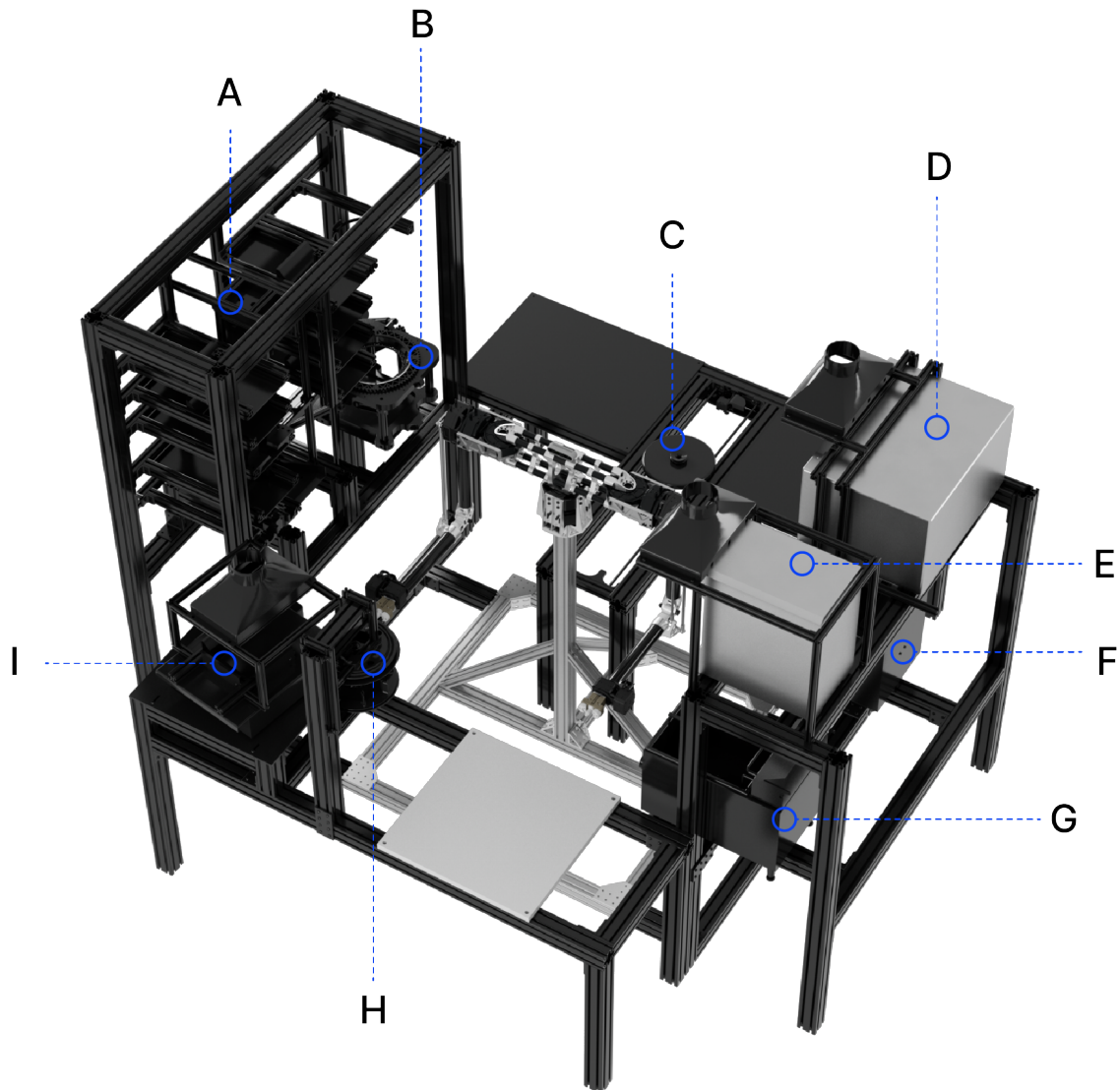


Figure 3.2: Layout of the cooking cell. The annotated components include: (A) storage shelf, (B) rotating pot, (C) dish trolley, (D) convection oven, (E) salamander broiler, (F) deep fryer, (G) pasta cooker, (H) spice dispenser, and (I) induction cooktop.

Figure 3.2 illustrates a structured cooking cell, comprised of multiple modular shelves and appliances arranged to utilize the available workspace efficiently. Each shelf, including the storage shelf (a), rotating pot (b), dish trolley (c), convection oven (d), salamander broiler (e), deep fryer (f), pasta cooker (g), spice dispenser (h), and induction cooktop (i), is designed as an independent module equipped with its own microcontroller for simplified operation and control. This modular structure allows for straightforward assembly and effortless reconfiguration, enabling rapid adaptation to various cooking tasks or menu requirements. By providing a well-defined and easily adjustable layout, the system reduces the complexity of collision-free motion planning and enhances the overall efficiency and productivity of the robotic cooking environment.

Figure 3.3 presents the suite of appliances and their associated tools that collectively form the YORI automated kitchen cell. Rather than relying on anthropomorphic robot hands to achieve dexterous manipulation, our system adopts a toolchanger-based approach, enabling the robot to seamlessly exchange specialized end-effectors as needed. This strategy alleviates the complexity of executing intricate manipulation motions and reduces the need for sophisticated, human-like grippers. Instead, functionality is integrated into the appliances themselves: the rotating pot and the dicing chamber, for instance, simplify cooking operations by handling tasks such as mixing and chopping directly within their own mechanisms. This approach not only enhances stability and reliability, but also supports scalability, making it easier to integrate additional appliances or tools as the system evolves. In doing so, we achieve a more robust and adaptable robotic cooking environment.

Figure 3.4 provides an overview of the integrated control architecture orchestrating the modular kitchen’s cooking processes. At the highest level, a centralized scheduler manages the operational flow by issuing commands and receiving status updates through shared memory. The appliance manager then sequentially relays these directives to individual microcontrollers, each identified by a unique ID, over a shared USB communication bus. By segmenting responsibilities across distinct modules, the system ensures that each appliance

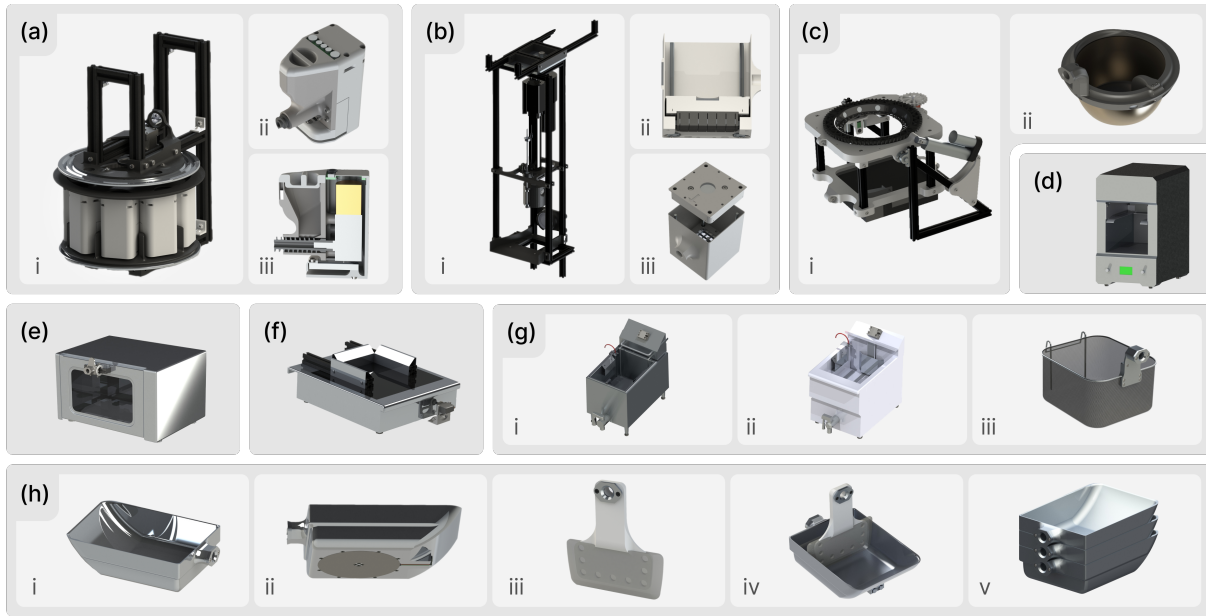


Figure 3.3: Overview of the integrated appliances and their corresponding tools forming the YORI automated kitchen cell. Depicted components include: (a) the spice dispenser with its ingredient dispensing modules, (b) the food processor and dicing chamber, (c) the rotating mixer with a detachable pot, (d) the salamander broiler, (e) the convection oven, (f) the induction cooktop, (g) the deep fryer and water boiler with their associated cooking basket, (h) the induction pan and its matching squeegee, and (i) a custom-designed induction pan paired with its specialized squeegee sweeping tool.

functions as an independent unit under a unified control framework. This modular approach simplifies coordination, streamlines communication, and ultimately enhances the overall reliability and scalability of the automated cooking environment.

Figure 3.5 illustrates the YORI System’s software architecture, emphasizing the importance of high-frequency communication for manipulator control. Critical operations—such as safety checks, state monitoring, and motor torque control—run at 1000 Hz to ensure rapid response times and system stability. Other components, including camera systems, task scheduling, and appliance management, operate at lower frequencies (30 Hz or 60 Hz)

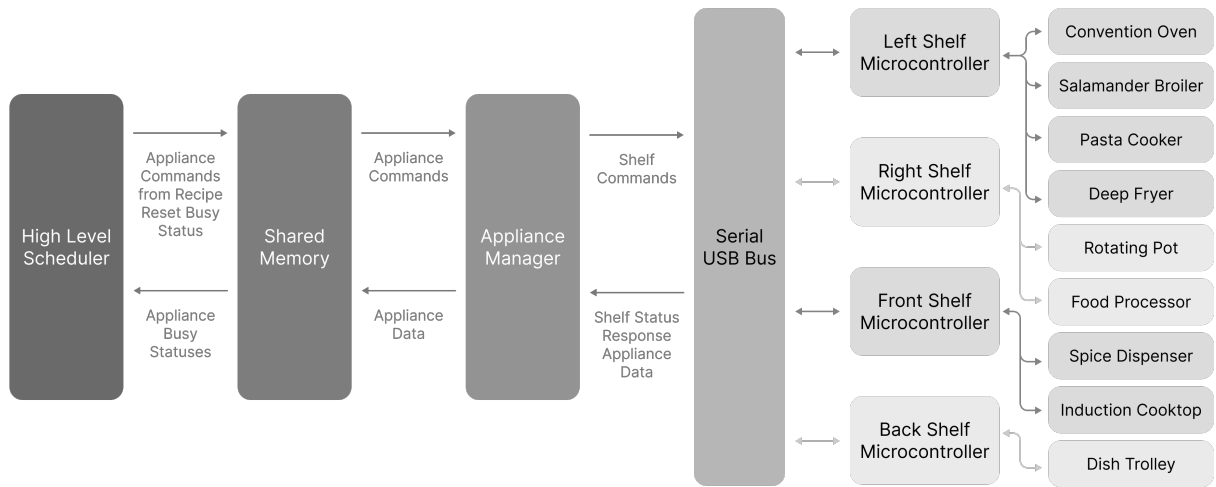


Figure 3.4: Flow diagram illustrating the control structure and messages used to control the integrated appliances in the kitchen cell. The High Level Scheduler communicates commands and statuses with the appliance manager through the systems shared memory. The appliance manager in turn communicates with each local microcontroller one at a time over the shared usb bus by sending commands tied to one of the unique microcontroller IDs.

without compromising overall efficiency. By employing shared memory for data exchange, the architecture minimizes latency and maximizes robustness, creating a solid foundation for more advanced manipulation strategies, including future implementations of force control and other sophisticated robotic capabilities.

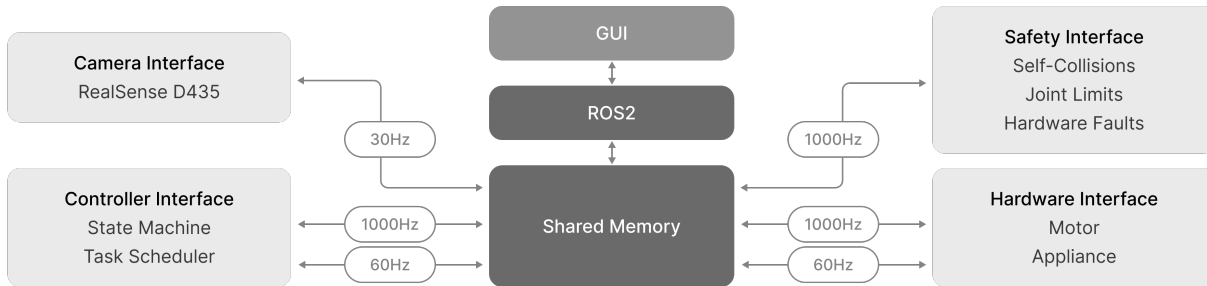


Figure 3.5: Overview of the YORI System’s software architecture, highlighting interfaces critical for high-frequency operations like safety checks, state checks, and motor control, communicating at 1000Hz. Conversely, components not requiring high-frequency interactions, such as camera systems, task scheduler, and appliance control, function at 30Hz or 60Hz. To guarantee swift communication and enhance system stability, data processing is executed via shared memory. Remote commands for the system are accepted through WiFi, leveraging the ROS2 middleware, solely upon specific order reception.

3.2 Implementation: Manipulator Control and Motion Planning

Manipulator Control

In the YORI system, cooking operations are categorized into two primary motion types: transporting and grasping. Transporting motions focus on relocating ingredients and tools, relying predominantly on the controller’s inverse dynamics capabilities. In contrast, grasping motions facilitate tool latching and interaction with the cooking cell environment and are governed by the impedance controller’s adjustable stiffness settings. By employing impedance control, each manipulator arm can smoothly transition between rigid and compliant states—remaining stiff when moving ingredients through the air and becoming flexible when manipulating tools or interacting closely with the environment as shown in Figure 3.6.

This functionality is achieved by dividing the dual-arm configuration into two kinematic chains, originating from either the torso or the shoulder yaw joint, depending on the specific

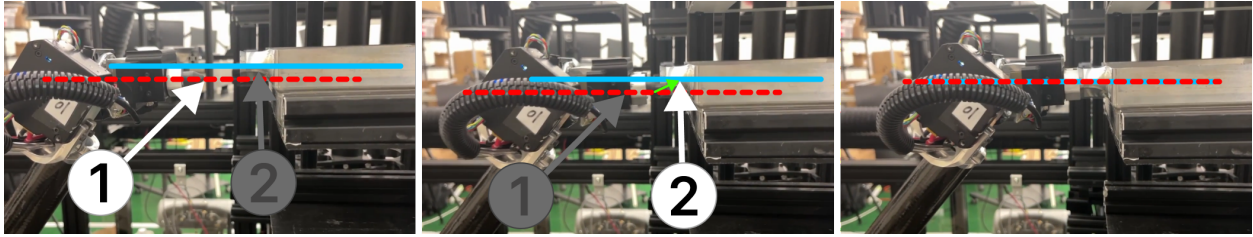


Figure 3.6: Example of impedance controller usage. (1) Current end-effector height and (2) Desired goal position height.

task. Each arm is managed by an independent controller, allowing for distinct stiffness values tailored to the requirements at hand. For instance, one arm may remain compliant while gently tossing fries for seasoning, while the other arm remains stiff to securely transport ingredients. Similarly, during two-arm tasks—such as sweeping a pan—the arm holding and moving the pan can maintain stiffness, while the arm gripping the squeegee remains compliant, consistently applying a downward force against the pan’s surface.

3.3 Summary

We successfully developed a dual-arm manipulator outfitted with proprioceptive actuators and implemented a practical control framework for its operation. This manipulator was seamlessly integrated into a cooking robot system, as illustrated in Figure 3.7, and demonstrated exceptional performance in both transporting and grasping tasks. Notably, in transporting motions, the system maintained incredibly reliable performance even when the payload changed during cooking, a distinction easily discernible from position control methods. This integration enabled us to streamline complex sensors and algorithms while still achieving advanced and reliable performance. The success of this project provides a robust foundation for future research into increasingly sophisticated and complex tasks.

In this chapter, we have introduced the hardware and software design processes of an autonomous robotic cooking system utilizing cutting-edge technology. This system’s develop-

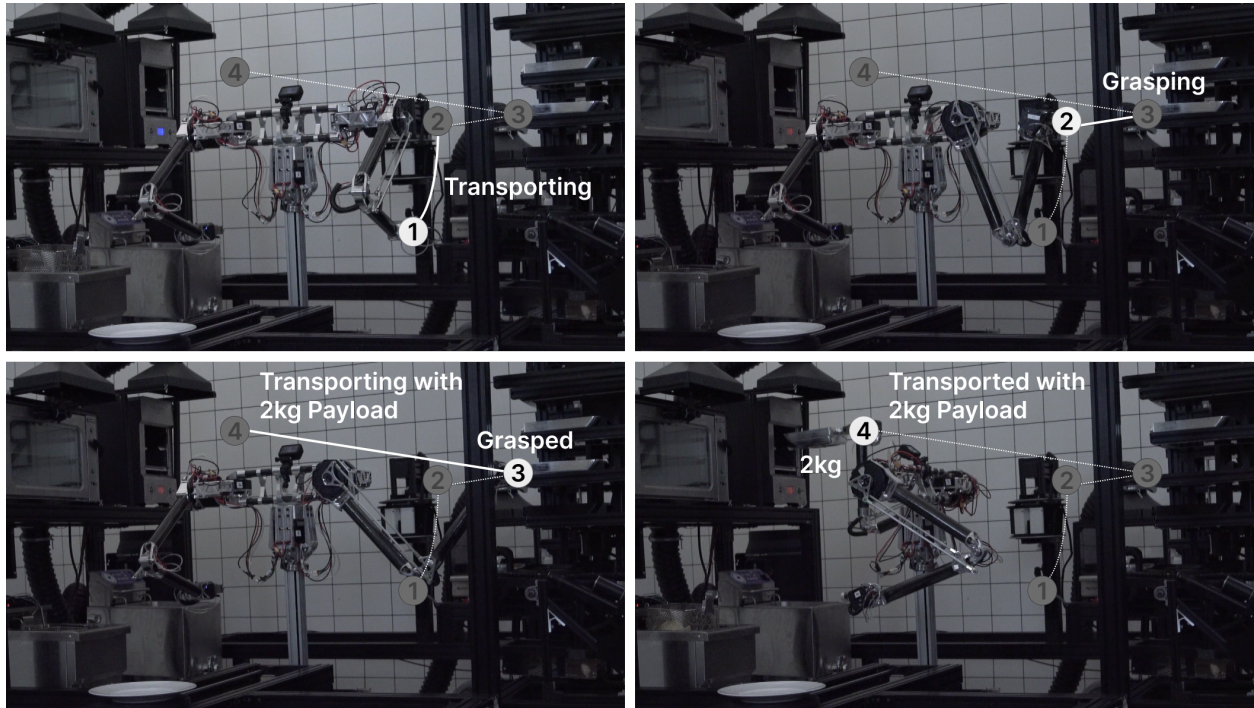


Figure 3.7: Example of transporting and grasping motions during the preparation of Steak Frites.

ment went beyond mere simulations and lab environments. The system was fully operational and showcased at a convention in Korea, shown in Figure 3.8. To enhance its reliability, we simplified the system, which allowed us to successfully demonstrate the continuous cooking of Steak Frites numerous times throughout the convention. The system's modularity and strategic simplifications enabled this achievement despite having only one day for preparation at the venue, underscoring the system's exceptional reliability and efficiency.



Figure 3.8: Demonstration of the autonomous robotic cooking system at a convention in Seoul, Korea, showcasing its ability to continuously cook Steak Frites.

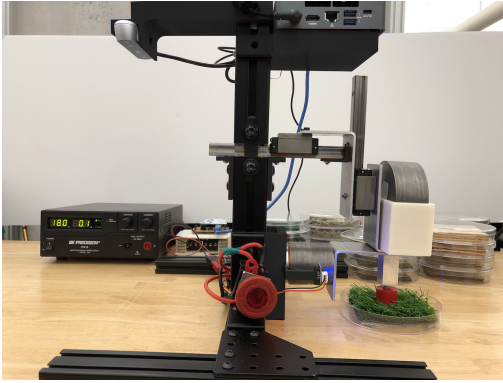
CHAPTER 4

Friction-Aware Grasping

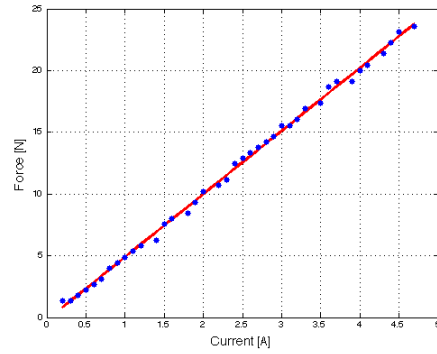
This chapter examines the integration of advanced control strategies using system property estimation techniques to enhance robotic grasping performance. Central to this investigation is the estimation of the friction coefficient, understood here as a system-level parameter arising from interactions between contacting surfaces rather than a property of a single material. The study utilizes the Surface Material Dataset for Robotics Applications (SMDRA) [NNA21], which provides RGB-D images and pixel-wise friction coefficient annotations, as a basis for training two deep learning models: UNet and TransUNet. These architectures were evaluated for their capability to estimate friction coefficients, thereby offering critical input for adjusting grasping forces.

Building upon these friction estimates, a control framework was developed to compute grasping forces that adapt dynamically in accordance with the end-effector’s orientation and the estimated friction coefficient. By incorporating indicator variables that consider the dominant face of an object, the framework ensures stable, slip-free manipulation while minimizing unnecessary torque exertion. The effectiveness of this friction-informed approach is validated through real-world scenarios, demonstrating its utility in addressing challenges associated with dynamic robotic manipulation tasks.

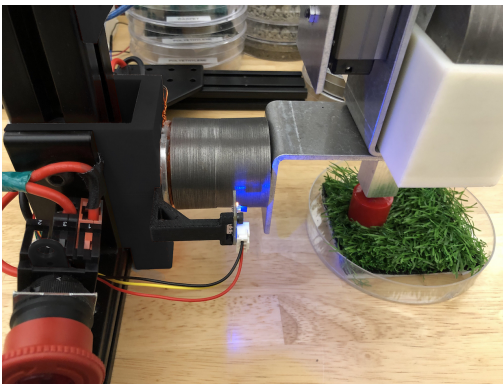
This research bridges the gap between theoretical system-level property estimation and practical robotics applications by integrating advanced computer vision methods with specialized control algorithms, thereby enabling safer, more efficient, and adaptable robotic systems that leverage friction coefficient estimations for refined, reliable grasping control.



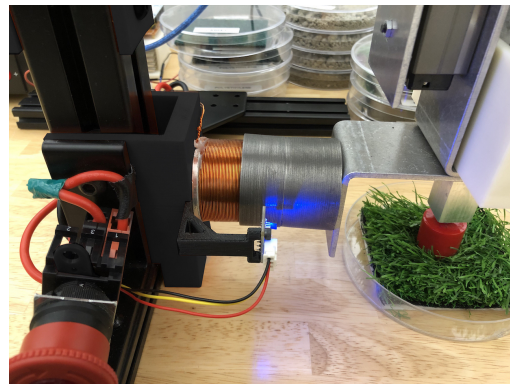
(a) Data collection setup.



(b) current [A] vs force [N].



(c) Before measurement.



(d) After measurement.

4.1 Dataset Preparation and Processing

This section details the preparation and validation of the Surface Material Dataset for Robotics Applications (SMDRA), encompassing the development of a novel friction measurement device, the collection of RGB-D and friction coefficient data, and subsequent validation using semantic segmentation models. The following sections describe the device preparation, data acquisition processes, and dataset verification procedures.

4.1.1 Friction Measurement Device

To facilitate accurate friction coefficient measurements across a range of materials, we developed a portable friction measurement device featuring a custom-built Voice Coil Motor

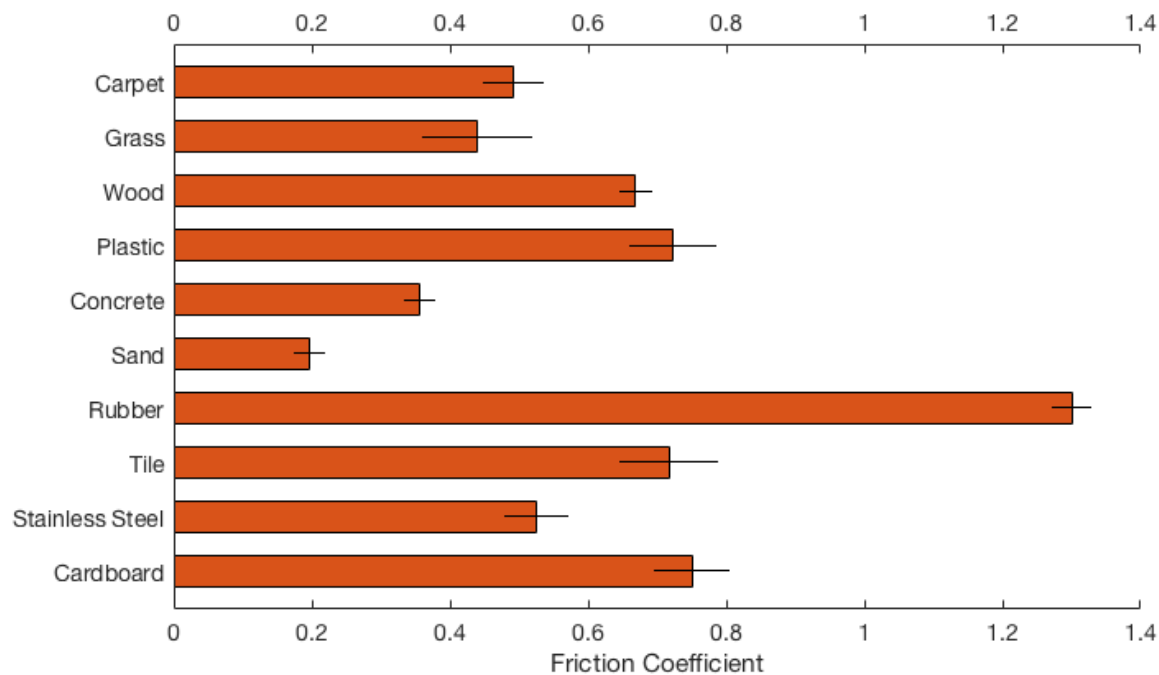


Figure 4.2: Measured friction coefficients for the materials in the SMDRA, including their mean values and standard deviations.

(VCM), a slider with adjustable mass, and linear rail guides. This configuration ensures consistent contact conditions, which are critical for obtaining reliable friction coefficient values in realistic settings.

As illustrated in Figure 4.1b, the VCM acts as a linear actuator generating force proportional to the applied current, calibrated with a load cell to ensure accuracy. By accommodating interchangeable contact materials (e.g., rubber, plastic) and supporting adjustable slider weights, the device enables a wide range of friction coefficient measurements up to approximately 1.3.

4.1.2 Data Collection Process

The SMDRA comprises RGB images, depth maps, and pixel-wise friction coefficient annotations for 10 materials commonly encountered in practical environments. The data collection

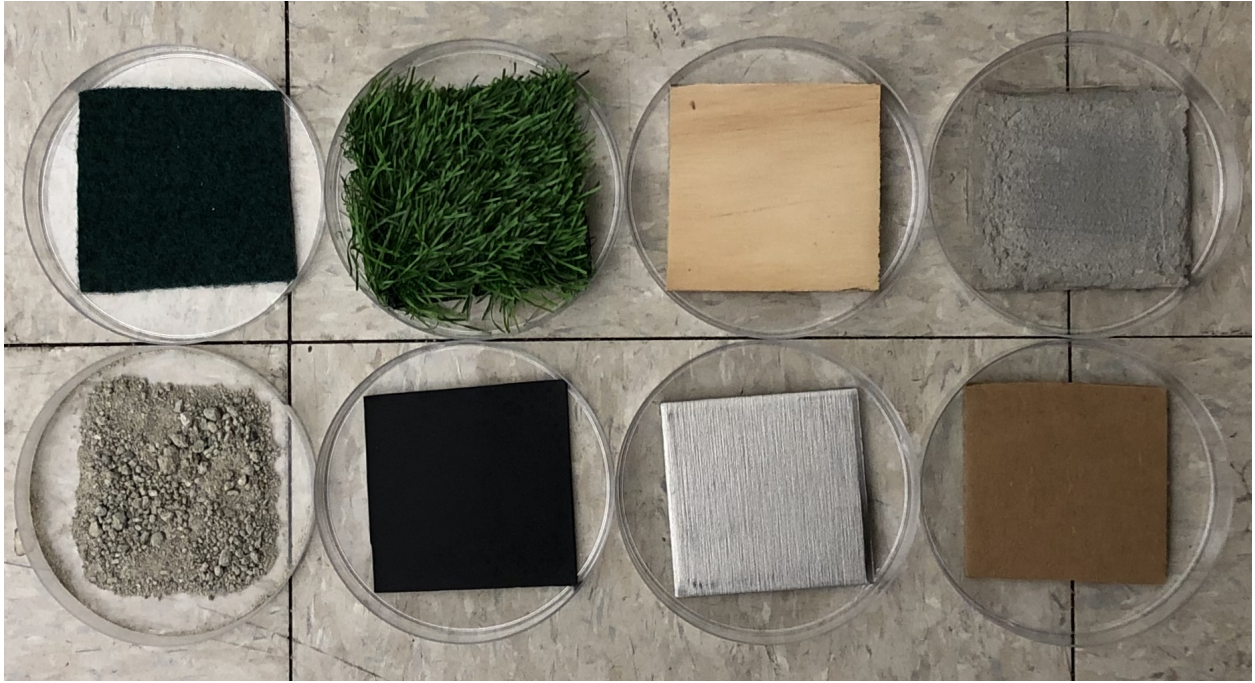


Figure 4.3: Example materials included in the dataset. Top row (from left):carpet, grass, wood, concrete; Bottom row (from left):sand, rubber, stainless steel, cardboard.

setup, shown in Figure 4.1a, integrates an Intel RealSense D415 stereo camera and the friction measurement device. Each material was arranged in petri dishes under controlled conditions to ensure consistent measurement environments.

For each material, friction coefficients were measured 100 times and the averaged values were annotated onto the corresponding RGB-D data. As shown in Figure 4.3, these materials span a broad spectrum of friction coefficients, supporting a range of applications in robotic manipulation tasks as shown in Figure 4.2. Throughout the measurement procedure, the device configurations before and after force application are illustrated in Figure 4.1c and Figure 4.1d. As the device initiates motion, a Hall sensor records the timing, allowing the instantaneous current to be measured. By leveraging the established linear relationship between current and force, and combining this force information with the known weight at the tip, the friction coefficient can be accurately determined.

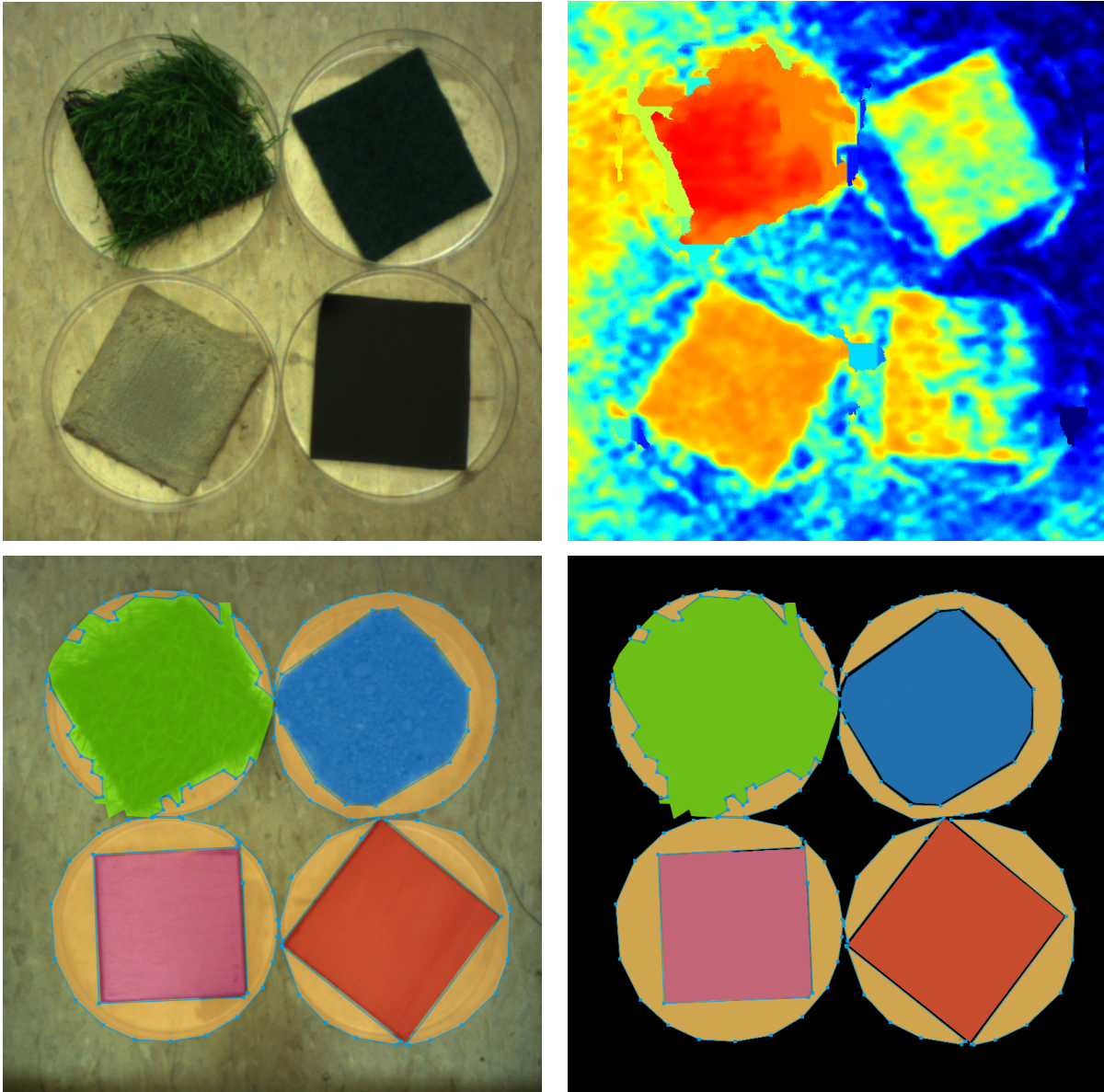


Figure 4.4: Examples of images captured by an active stereo camera and annotated using the COCO Annotator tool. The top row shows an RGB image (left) and a depth image (right) of a sample containing carpet, grass, concrete, rubber, plastic, and tile. The bottom row presents another RGB (left) and annotated depth image (right) of a sample composed of carpet, grass, stainless steel, wood, plastic, and tile.

4.1.3 Validation of Dataset Utility

To validate the utility of the SMDRA, we trained two semantic segmentation models—U-Net and Fully Convolutional Network (FCN)—on both RGB and RGB-D inputs. Table 4.1 summarizes the training parameters used to ensure a fair comparison between the two architectures. The results, presented in Table 4.2, highlight the dataset’s overall quality and demonstrate that incorporating depth information (RGB-D) consistently enhances segmentation performance. Representative segmentation outputs are shown in Figure 4.5.

Although these segmentation results are encouraging, they primarily serve as a preliminary verification step, treating the problem as a segmentation (classification) task rather than directly addressing the friction estimating problem. By confirming the dataset’s integrity and the reliability of its annotations, we establish a solid foundation upon which more sophisticated regression-based approaches—such as those involving friction coefficient estimation and friction-informed grasp planning—can be developed. In this way, the SMDRA paves the way for future advancements in friction-aware robotic manipulation strategies.

Table 4.1: Training Parameters for Semantic Segmentation Models

Parameter	U-Net	FCN
Input Shape	$512 \times 512 \times 4$	$512 \times 512 \times 4$
Batch Size	4	4
Optimizer	Adam	Adam
Activation Function	LReLU	LReLU
Epochs	25	25

Table 4.2: Performance Metrics for Semantic Segmentation Models

Model	Pixel Accuracy	Mean Accuracy	Mean IU
FCN (RGB)	95.3%	90.1%	79.8%
FCN (RGB-D)	97.3%	95.5%	89.7%
U-Net (RGB)	95.6%	86.2%	75.6%
U-Net (RGB-D)	96.3%	94.7%	91.3%

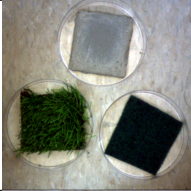
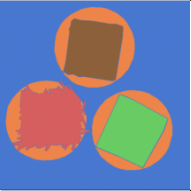
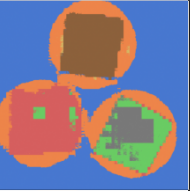
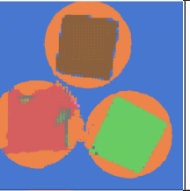
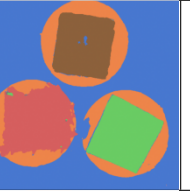
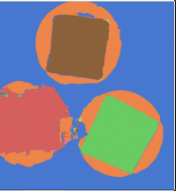

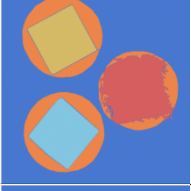
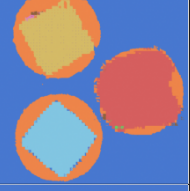
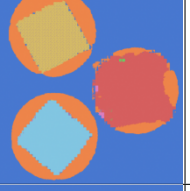

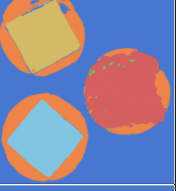
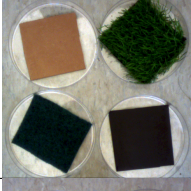
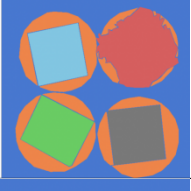


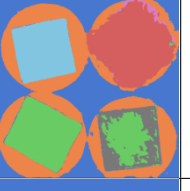
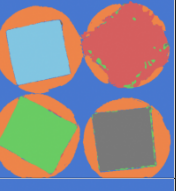

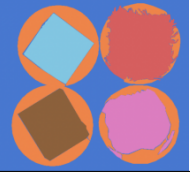
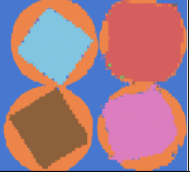
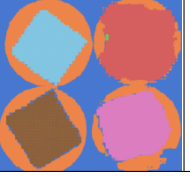
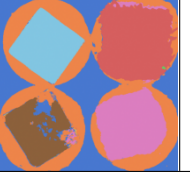
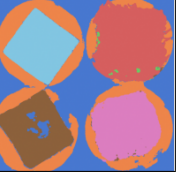
		FCN-8s		U-Net	
Raw Image	Ground Truth	RGB	RGB-D	RGB	RGB-D
					
					
					
					

Figure 4.5: Segmentation results for four example test cases. The materials involved include grass, concrete, rubber, plastic, tile, stainless steel, cardboard, carpet, and sand. Each column compares the raw image, ground truth, and segmentation outputs from FCN and U-Net using both RGB and RGB-D inputs.

4.2 Model Architecture and Training

This section outlines the architectures and training procedures employed for estimating the friction coefficient using regression models. Two distinct neural network architectures were evaluated: UNet and TransUNet. Each model was designed and trained to predict the friction coefficient from input images, leveraging their unique strengths in handling spatial and contextual information.

UNet Architecture

The UNet architecture, widely recognized for its effectiveness in segmentation tasks, was adapted for the regression problem of friction coefficient estimation. The network consists of an encoder-decoder structure with skip connections that preserve spatial details across scales. The encoder extracts hierarchical features, while the decoder reconstructs these into friction coefficient estimates.

Key parameters for the UNet implementation:

- **Input size:** 512x512
- **Loss function:** $0.2 \times \text{MSE} + 0.8 \times \text{MAE}$
- **Optimizer:** Adam with 0.0001 Learning Rate.
- **Batch size:** 8
- **Number of epochs:** 30

TransUNet Architecture

TransUNet combines a Transformer encoder with the UNet's convolutional structure, allowing it to capture global dependencies while preserving spatial precision. This hybrid

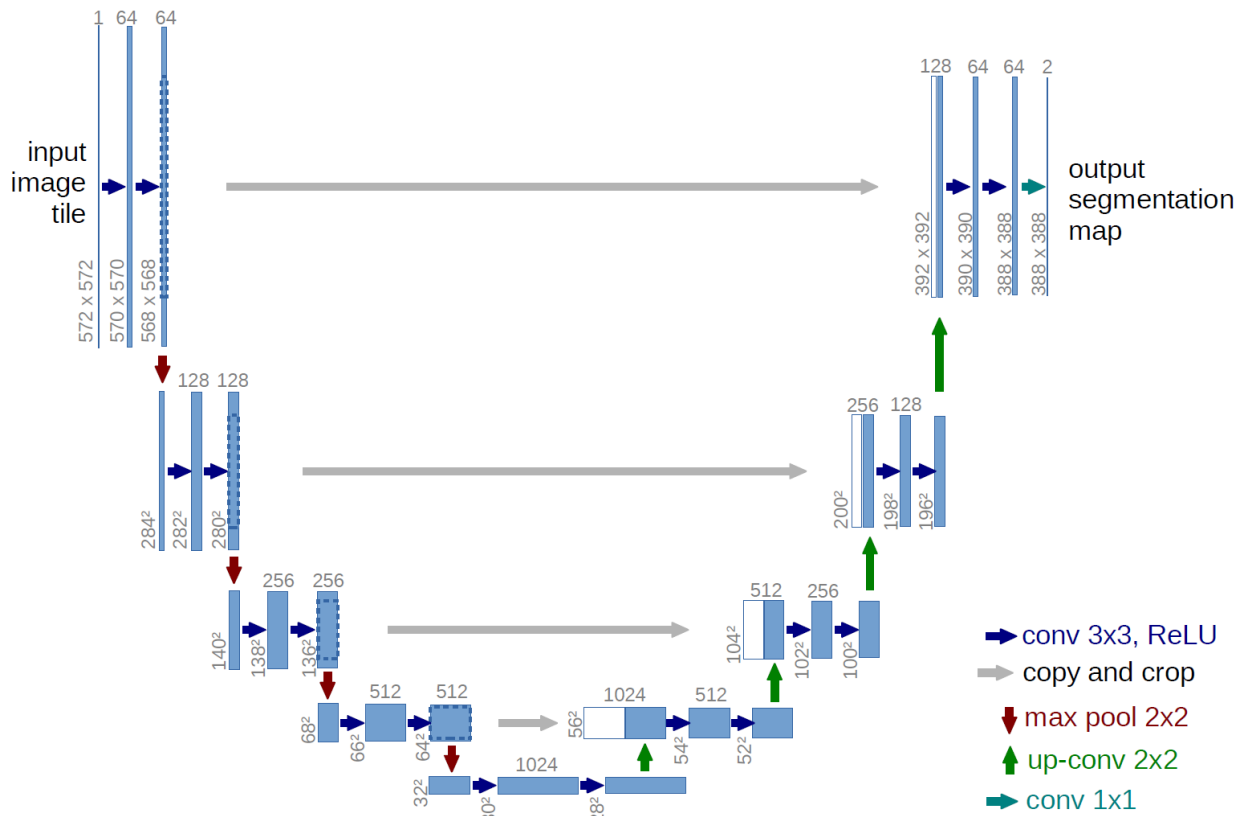


Figure 4.6: UNet architecture [RFB15].

architecture is particularly well-suited for tasks requiring contextual understanding alongside detailed reconstruction, making it an ideal choice for friction coefficient estimation.

Key parameters for the TransUNet implementation:

- **Input size:** 512x512
- **Number of attention heads:** 4
- **Loss function:** $0.2 \times \text{MSE} + 0.8 \times \text{MAE}$
- **Optimizer:** Adam with 0.0001
- **Batch size:** 8
- **Number of epochs:** 100

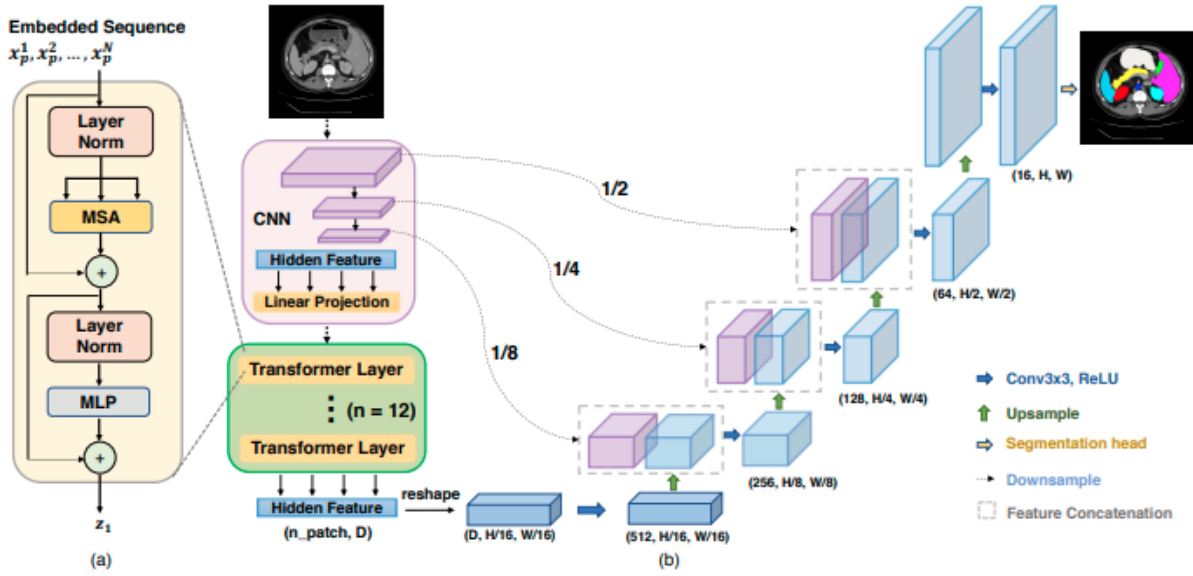


Figure 4.7: TransUNet architecture combining Transformer encoding and convolutional decoding [CLY21].

Training Procedure

Each model was trained on a dataset of images paired with ground truth friction coefficients. The training dataset consisted of 200 samples, divided into training, validation, and test sets in a ratio of 8:1:1. Data augmentation techniques, including Brightness adjustment, Rotation, were applied to improve generalization.

The performance of each architecture will be compared to assess their strengths and limitations in predicting friction coefficients, with detailed results presented in the following sections.

4.3 Training Results

As shown in Table 4.3, the TransUNet significantly outperforms the U-Net, achieving a Mean MAE of 0.13 compared to the U-Net’s 0.24. Although the U-Net’s performance is

Table 4.3: Mean MAE Results for U-Net and TransUNet

Model	Mean MAE
U-Net	0.24
TransUNet	0.13

comparatively weaker, it still achieves a Mean MAE below 0.25, suggesting the potential for further improvement. The U-Net’s ability to capture rich local features is evident, as it effectively delineates object boundaries. This observation is supported by the example output in Figure 4.8, which demonstrates its strength in boundary recognition tasks.

In contrast, Figure 4.9 highlights TransUNet’s superior performance in estimating friction coefficients despite slightly less pronounced boundaries. Given that friction coefficients are system-level properties, TransUNet’s more vigorous global context modeling appears to play a pivotal role in its accuracy. This result implies that while the U-Net excels at local detail, the TransUNet’s emphasis on broader contextual information leads to more precise friction estimation.

While TransUNet’s scalability and generalization capacity remain open questions—particularly given the current dataset’s limited size—these results confirm that friction coefficients can be effectively estimated. As the amount and variety of training data increase, performance will likely improve further. Under such circumstances, TransUNet’s ability to integrate global context may become even more advantageous, potentially extending its lead over U-Net and significantly advancing friction-aware robotic manipulation.

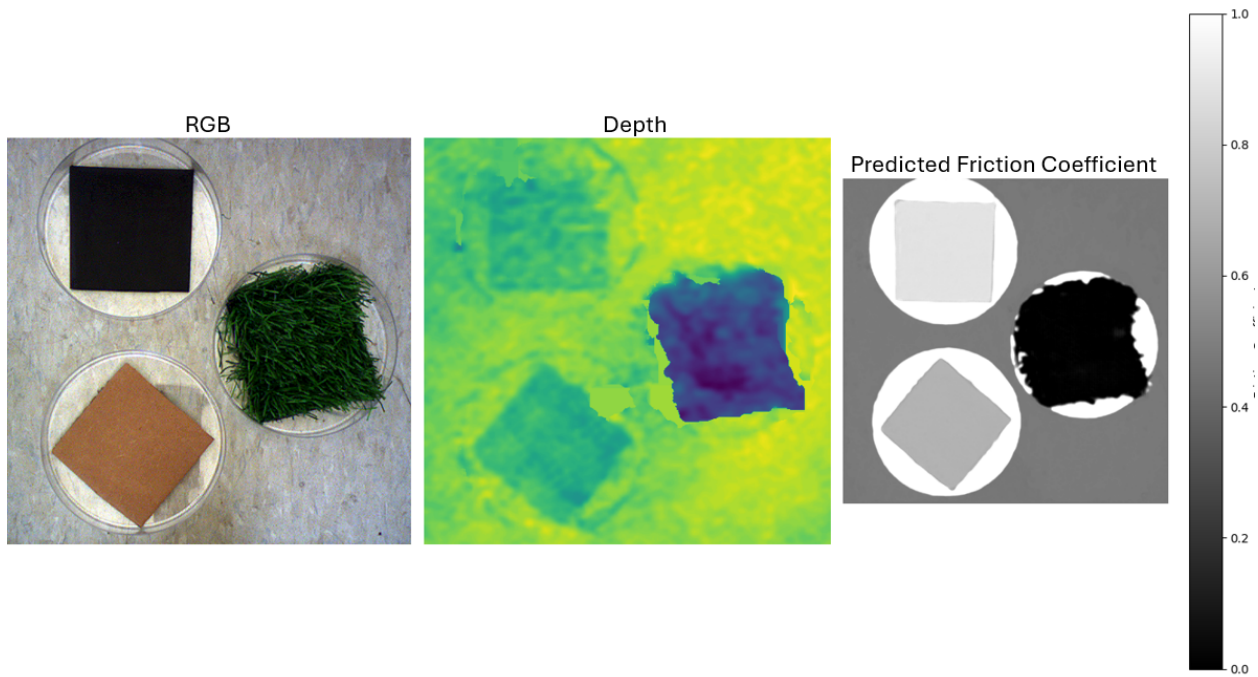


Figure 4.8: Representative U-Net output demonstrating strong boundary delineation and effective local feature recognition.

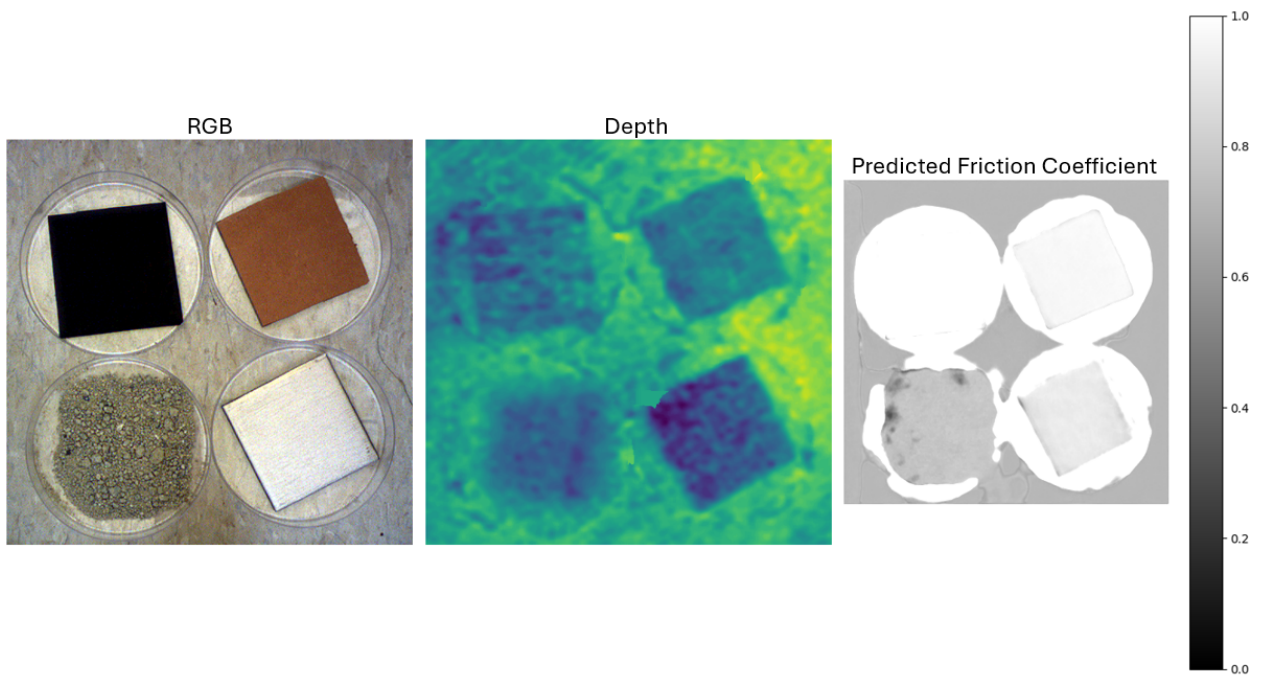


Figure 4.9: Representative TransUNet output illustrating improved friction coefficient estimation, likely due to enhanced global context modeling.

4.4 Friction-Aware Grasping

After estimating friction coefficients with the trained model, we integrated these values into a friction-aware grasping strategy. Instead of relying solely on position control, our approach continuously adjusts the grasping force using real-time IMU measurements of the gripper’s orientation and the initially estimated friction coefficient. For simplicity, the internal dynamics of the gripper were not considered, and the object’s weight was assumed to be known. By incorporating gravitational effects via IMU data, the effective load on the object could be computed accurately, allowing the grasping forces to adapt as the orientation changes and thereby ensuring stable and efficient manipulation.

4.4.1 Friction-Aware Grasping Algorithm

The algorithm leverages the friction coefficient (μ) to determine the required grasping forces on the gripper’s left and right fingers. Indicator variables (δ_{left} and δ_{right}) identify the object’s dominant face relative to the gripper, ensuring that the computed forces reflect both local frictional properties and global orientation-dependent loading.

Variables

- μ : Coefficient of friction between the gripper and the object.
- $F_{z,i}$: Negative z -component force acting on face i , where $i \in \{\text{left, right, top, bottom, front, back}\}$.
- F_{load} : Load acting on the dominant gripper finger (either left or right).
- $F_{z,\text{sum}}$: Sum of all z -component forces acting on the object.
- $\delta_{\text{left}}, \delta_{\text{right}}$: Indicator variables:

$$\delta_{\text{left}} = \begin{cases} 1, & \text{if the dominant face is “left”,} \\ 0, & \text{otherwise,} \end{cases}$$

$$\delta_{\text{right}} = \begin{cases} 1, & \text{if the dominant face is "right",} \\ 0, & \text{otherwise.} \end{cases}$$

Algorithm Steps

1. Compute the total z -component force acting on the object:

$$F_{z,\text{sum}} = \sum_{i=1}^N F_{z,i}$$

where N represents the number of faces contributing to the force (e.g., top, bottom, front, back).

2. Calculate the net force acting on the object:

$$F_{z,\text{net}} = F_{z,\text{sum}} + F_{\text{load}}$$

3. Determine the grasping forces:

$$f_{\text{left}} = \delta_{\text{left}} \cdot \max\left(\left|\frac{F_{z,\text{net}}}{2\mu}\right|, F_{\text{load}}\right) + (1 - \delta_{\text{left}}) \cdot \left|\frac{F_{z,\text{net}}}{2\mu}\right|$$

$$f_{\text{right}} = \delta_{\text{right}} \cdot \max\left(\left|\frac{F_{z,\text{net}}}{2\mu}\right|, F_{\text{load}}\right) + (1 - \delta_{\text{right}}) \cdot \left|\frac{F_{z,\text{net}}}{2\mu}\right|$$

4. Multiply the resulting torque values by a safety factor of 3 to ensure reliable, robust grasping.
5. Output the computed grasping forces $(f_{\text{left}}, f_{\text{right}})$.

This formulation integrates dominant face selection via indicator variables, ensuring that applied grasping forces continuously and intelligently adjust according to the object's frictional properties and orientation.

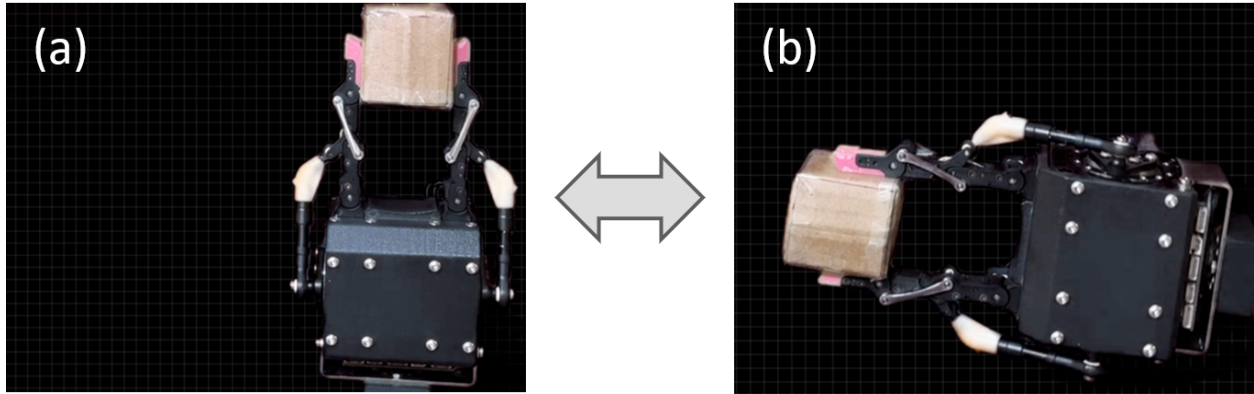


Figure 4.10: Illustration of the friction-aware grasping experiment setup. In (a), the gripper is shown before rotation, and (b) illustrates a rapid rotation scenario. This dynamic testing environment evaluates the gripper’s ability to adapt its grasping forces in real-time to changes in object orientation and the estimated friction coefficient.

4.4.2 Experimental Validation

To evaluate the proposed friction-aware grasping strategy, we established a dynamic experimental setup, as illustrated in Figure 4.10. In this environment, the gripper can rapidly rotate and change its orientation. This allows us to assess how well it adapts its grasping forces to varying gravitational effects utilizing the estimated friction coefficient. By doing so, the experiment tests the system’s ability to maintain stable, efficient manipulation even when subjected to sudden orientation changes and load distributions.

Figure 4.11 compares torque profiles under traditional position-based control and the proposed friction-aware force-based approach, tested with both 100g and 200g boxes. Even after applying a safety factor of three to the computed torque values, the friction-aware method significantly reduces both peak and overall torque compared to fixed-grasp, position-controlled scenarios. This improvement in efficiency is evident in the substantially lower torque profiles observed during the experiment.

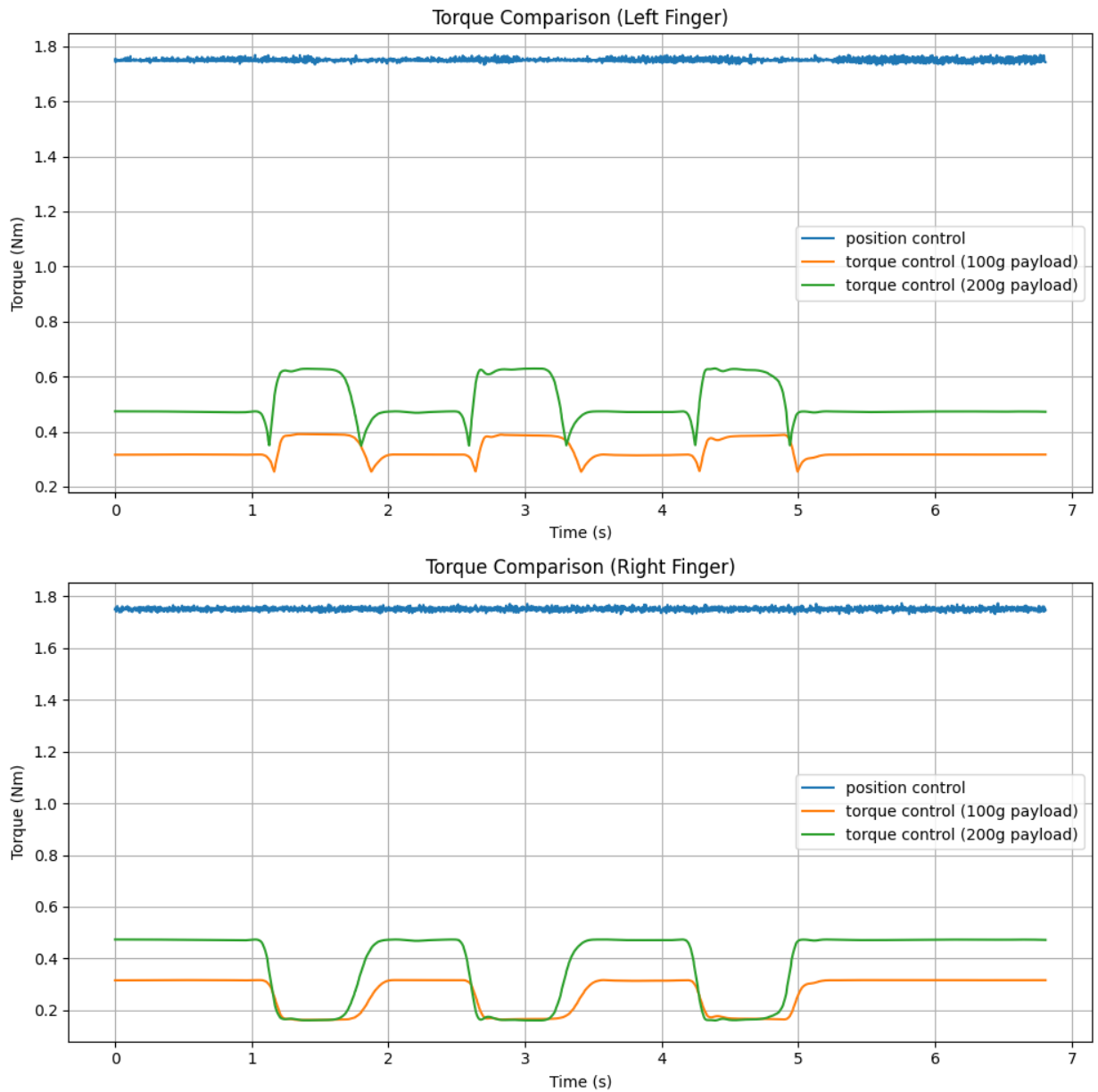


Figure 4.11: Comparison of torque profiles under traditional position-based control and friction-aware force-based control for both 100g and 200g boxes. Even with a safety factor of 3 applied, the friction-aware approach significantly reduces both peak and overall torque, demonstrating reliable grasping performance across varying loads.

4.5 Summary

This chapter has demonstrated how accurately estimating system properties, specifically friction coefficients, can significantly enhance robotic grasping and manipulation. The proposed methodology integrates advanced deep learning architectures (UNet, TransUNet) with a control framework validated using the SMDRA dataset. These integrated approaches enable precise force distribution and reduced torque requirements, all while maintaining grasp stability.

By incorporating estimated friction coefficients into force computation, the system intelligently modulates grasping forces in response to changing object orientations and weight distributions. This friction-aware strategy not only ensures stable and efficient manipulation but also decreases mechanical stress on the manipulator and provides a robust safety margin. The resulting improvements in operational efficiency and adaptability are particularly valuable in unstructured, dynamic environments.

This work bridges the gap between data-driven system property estimation and practical manipulation strategies, paving the way for safer, more reliable, and more scalable robotic systems. As larger datasets and more accurate estimation models emerge, the presented friction-aware approach is poised to further advance the capabilities of robots operating in both human-centric and industrial settings.

CHAPTER 5

Adaptive Video Imitation for Dynamic Manipulation

5.1 Overview

Dynamic manipulation remains one of the most challenging areas in robotics, requiring the execution of tasks involving rapid and complex motion changes while interacting with objects of varying properties. The inherent nonlinearities and uncertainties present in such scenarios pose significant difficulties for state estimation and control. Inspired by the human capability to perform dynamic tasks, such as throwing or catching, robotics research has sought to narrow this gap through both optimization-based and learning-driven approaches [HS05, HCW23, KK23, LWZ23, ZLT23, ZSL20].

This chapter introduces **AVID** (Adaptable Video Imitation for Dynamic Manipulation), a framework currently under development that aims to advance existing methodologies for dynamic manipulation. In contrast to approaches relying primarily on video motion re-targeting, AVID leverages information extracted from human demonstrations to inform a robot dynamics model. By integrating kinematic adaptation with explicit consideration of the robot’s dynamic properties, AVID aspires to enable more energy-efficient and stable manipulations—even as object conditions and orientations vary.

The anticipated contributions of this work include:

- **Human-to-Robot Dynamic Adaptation:** Rather than limiting itself to simple motion re-targeting, AVID analyzes human demonstration videos and incorporates the resulting insights into a robot’s dynamic model. This approach aims to generate control

policies that are dynamically feasible and robust to variations in object properties and task conditions.

- **Energy-Efficient Dynamic Manipulation:** Drawing inspiration from human strategies, AVID seeks to reduce overall energy consumption and lower peak torque requirements in dynamic tasks. By more fully exploiting the robot’s inherent dynamics, this method could surpass conventional position- or trajectory-based control approaches.
- **Latent Plan Transformer (LPT)-Driven Adaptability:** By employing the Latent Plan Transformer (LPT) [KXZ24], AVID endeavors to achieve greater adaptability and robustness. The LPT’s ability to abstract latent plans from demonstration data may yield efficient trajectory generation, improved inference speed, and enhanced performance across diverse scenarios.

5.1.1 Box Receiving as a Representative Dynamic Manipulation Task

Dynamic manipulation tasks often demand that robots handle rapidly moving objects in unpredictable environments, requiring swift and robust responses under uncertainty. Among various possibilities, we selected **box receiving** as a representative challenge. Successfully receiving a thrown box necessitates rapid arm movements, precise timing, and careful interaction control—characteristics that encapsulate many core elements of dynamic manipulation. Moreover, such tasks highlight the advantages of our proprioceptive manipulator, whose intrinsic sensing and control capabilities offer a means to adapt in real-time. The robot must continuously adjust its motion plan as the box’s trajectory evolves, making both sequence execution and replanning critical.

While inherently complex, the box-receiving task can be simplified to three pitch joints at the shoulder, elbow, and wrist, mirroring the human arm’s configuration for this specific interaction. This simplification preserves essential dynamic features—such as high-speed motion, effective momentum management, and intricate contact handling—while ensuring

that the problem remains tractable. Consequently, the box-receiving task provides an accessible yet sufficiently demanding testbed for exploring and validating strategies in dynamic manipulation.

Building on this problem definition, the following sections introduce our approach to translating human demonstration data into robot dynamics and selecting appropriate control strategies. This groundwork aims to guide the robot toward stable, efficient, and adaptable performance in dynamic manipulation tasks.

5.1.2 Our Approach: Translating Human Demonstrations into Robot Dynamics

Once human pose estimation is obtained, the human arm’s joint angles, specifically at the shoulder, elbow, and wrist pitch joints, can be extracted, which is crucial for the box-receiving scenario. By comparing the ratios of human and robot arm lengths, the box size and position can be appropriately scaled and adjusted. This process allows the robot’s end-effector to be positioned relative to the adjusted box coordinates, thereby maintaining the spatial relationships observed in the original human demonstration.

Next, the robot’s wrist, elbow, and shoulder joint angles are determined to replicate the human arm configuration as closely as possible. The differences in joint positions between consecutive video frames, combined with the known frame rate, facilitate the calculation of joint velocities and accelerations. By subsequently applying the inverse dynamics of the robot model, the required joint torques can be derived. The robot’s dynamics are described by:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + J^\top(q)f_{\text{ext}}, \quad (5.1)$$

where $M(q)$ is the inertia matrix, $C(q, \dot{q})$ is the Coriolis and centrifugal matrix, $G(q)$ is the gravity vector, τ denotes the joint torques, and f_{ext} represents external forces.

5.1.3 Control Strategies

The choice of control method depends on the policy’s performance and stability, leading to three potential scenarios:

1. **Tracking Trajectories: Inverse Dynamics Controller**

If precise trajectory tracking is paramount, an inverse dynamics controller can compute the required joint torques to follow the desired trajectory accurately.

2. **Compliant Interaction: Impedance Controller**

When compliant interaction between the robot and the box is essential, an impedance controller can modulate the interaction dynamics to ensure safe, adaptable behavior.

3. **Unstable Trajectories: Model Predictive Control (MPC)**

If the policy yields unstable trajectories yet provides valuable directional information, a Model Predictive Control (MPC) approach can refine the trajectory based on the policy’s output, improving overall stability and performance.

Although the AVID framework remains a work in progress, these directions underscore its potential to significantly enhance robotic systems’ capabilities in dynamic manipulation. By integrating human-inspired insights, leveraging the robot’s inherent dynamic model, and employing flexible control strategies, AVID aims to pave the way toward safer, more efficient, and more adaptable robotic manipulation in complex environments.

5.2 Dataset Preparation

5.2.1 Data Collection

The dataset was collected to evaluate the performance of dynamic motion control using robotic manipulators, informed by 2D whole-body pose estimation and subsequent inverse

dynamics calculations. Data were gathered across multiple conditions and box configurations to ensure robustness and generalizability.

Experimental Setup

- **Camera:** ZED2 RGB-D stereo camera.
- **Resolution:** 672×376 pixels.
- **Frame Rate:** 100 Hz.
- **Environment:** Controlled laboratory environment with consistent lighting and minimal occlusions.
- **Box Types:** Five distinct box types, each with two weight conditions:
 - **Empty box:** No internal load.
 - **Weighted box:** Containing a 1 kg weight.
- **Motion Types:** For each box configuration, two motion types were demonstrated:
 1. **Non-impact mitigation motion:** Unmodified dynamic manipulation without strategies to mitigate external impacts.
 2. **Impact mitigation motion:** Dynamic manipulation with explicit control strategies to minimize impact forces.
- **Demonstrations:** Each combination of box type, weight, and motion type was performed five times, resulting in

$$5 \text{ box types} \times 2 \text{ weights} \times 2 \text{ motion types} \times 5 \text{ trials} = 100 \text{ demonstrations.}$$

RGB data were collected throughout each trial, ensuring the availability of both color and depth information for analysis.

5.2.2 Data Processing

The collected data underwent several processing stages to extract meaningful features for dynamic motion analysis. These steps are described below: [Con20]

Pose Estimation and Keypoint Filtering

1. **2D Pose Estimation:** Using **MMPose** with the **RTMPose** model, we obtained 2D keypoints for the entire human skeleton.
 - **Backbone:** `rtmw-x_8xb320-270e_cocktail14-384x288`
 - **Weights:** `rtmw-x_simcc-cocktail14_pt-ucoco_270e-384x288-f840f204_20231122.pth`
2. **Keypoint Filtering:** A low-pass filter was applied to each joint's 2D trajectory, smoothing out noise and ensuring stable keypoint sequences.
3. **Skeleton Simplification:** We selected a subset of keypoints (right hip, shoulder, elbow, wrist, and finger) for subsequent analysis. This reduced skeleton was visualized to confirm proper filtering and extraction.

Joint Dynamics and Inverse Dynamics

1. **Calculation of Joint Angles:** Stabilized keypoints provide the necessary information to determine human arm lengths and proportions. By comparing these measurements with the robot's known dimensions, a scaling ratio is established and applied to both the box's position and its size. Once these scaled parameters are in place, the fingertip's relative position and the wrist joint position with respect to the adjusted box configuration can be derived. Subsequently, the remaining joint positions are computed to maintain a posture that closely aligns with the original human demonstration. This approach ensures that the robot's configuration accurately reflects the observed motion

while effectively accommodating differences in limb length and overall geometry.

2. **Joint Velocities and Accelerations:** First-order and second-order numerical derivatives of the joint angles were calculated to obtain joint velocities and accelerations.
3. **Inverse Dynamics:** The calculated joint angles, velocities, and accelerations were used as inputs to an inverse dynamics model of the target robotic manipulator.
 - **Inverse Dynamics Model:** The computation considered the kinematic and dynamic properties of the robot, including mass distribution and joint constraints.

Visualization and Analysis

The processed data were visualized to validate the pipeline and analyze the results. The following plots were generated:

1. **Human and Box Pose Estimation:** RGB image overlaid with detected skeletons and the box, providing qualitative validation of the pose estimation and scaling process.
2. **Keypoint Stability:** Raw and filtered keypoint trajectories for **right hip, right shoulder, right elbow, right wrist, and right finger**.
3. **Simplified Skeleton:** Visualizations of the simplified skeleton for dynamic analysis.
4. **Joint Dynamics:** Plots of joint angles, velocities, and accelerations for each joint in the simplified skeleton.
5. **Torque Profiles:** Estimated torque requirements for the manipulator to replicate observed motions under both motion types and box conditions.

This comprehensive processing pipeline ensured reliable data for analyzing dynamic manipulation, enabling insights into the control strategies required for efficient and adaptive robotic manipulation.

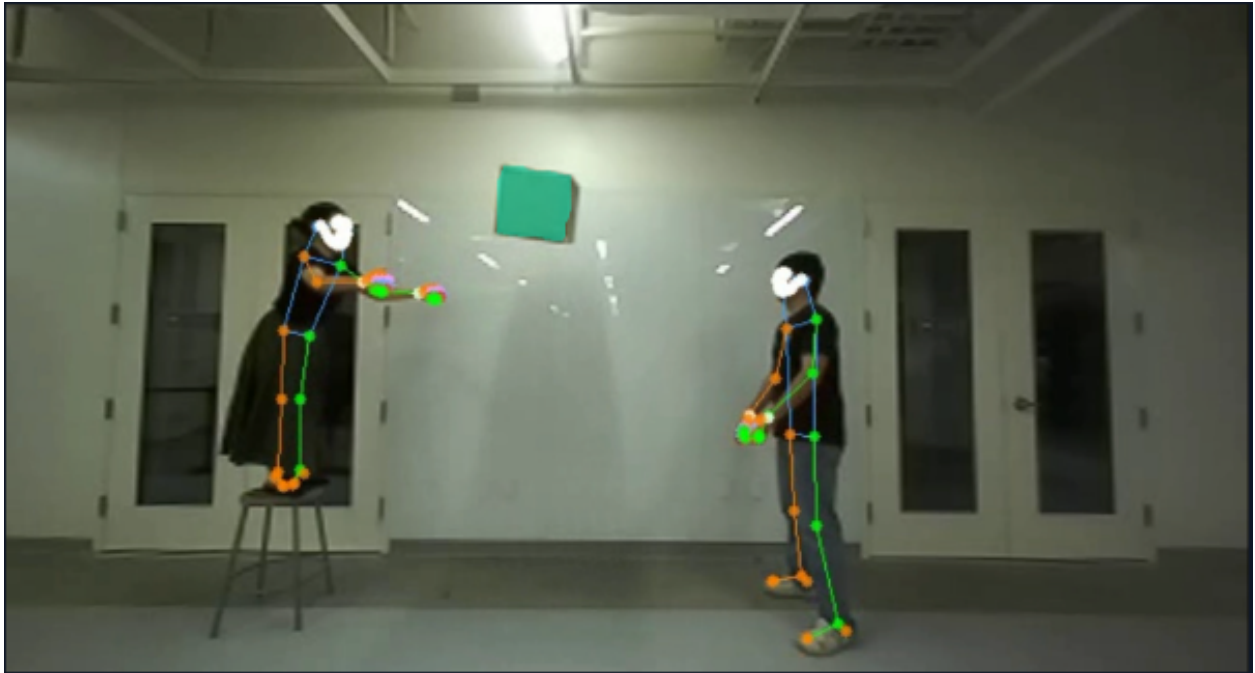
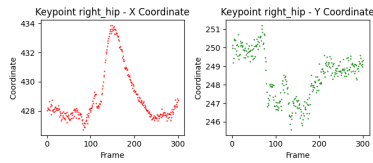
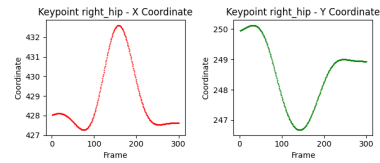


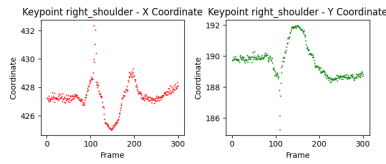
Figure 5.1: Pose estimation, and box segmentation results are overlaid on an RGB image. The detected keypoints demonstrate accurate alignment with the subject's body structure.



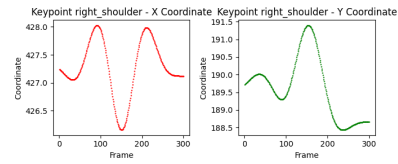
(a) Raw trajectory: Right Hip



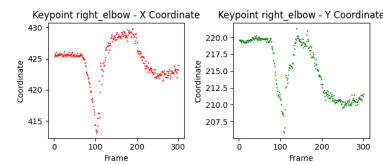
(b) Filtered trajectory: Right Hip



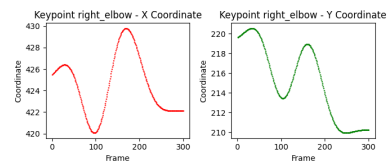
(c) Raw trajectory: Right Shoulder



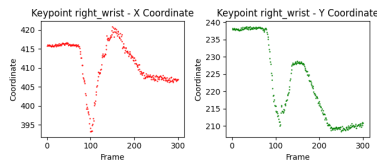
(d) Filtered trajectory: Right Shoulder



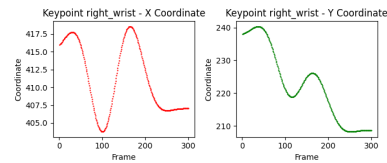
(e) Raw trajectory: Right Elbow



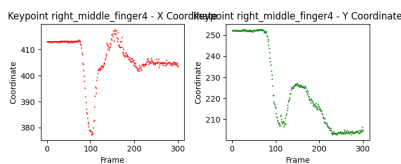
(f) Filtered trajectory: Right Elbow



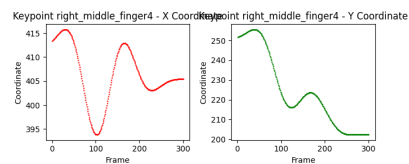
(g) Raw trajectory: Right Wrist



(h) Filtered trajectory: Right Wrist



(i) Raw trajectory: Right Finger



(j) Filtered trajectory: Right Finger

Figure 5.2: Keypoint trajectories for the right-side joints (hip, shoulder, elbow, wrist, and finger). Each joint includes raw and filtered trajectories for comparison.

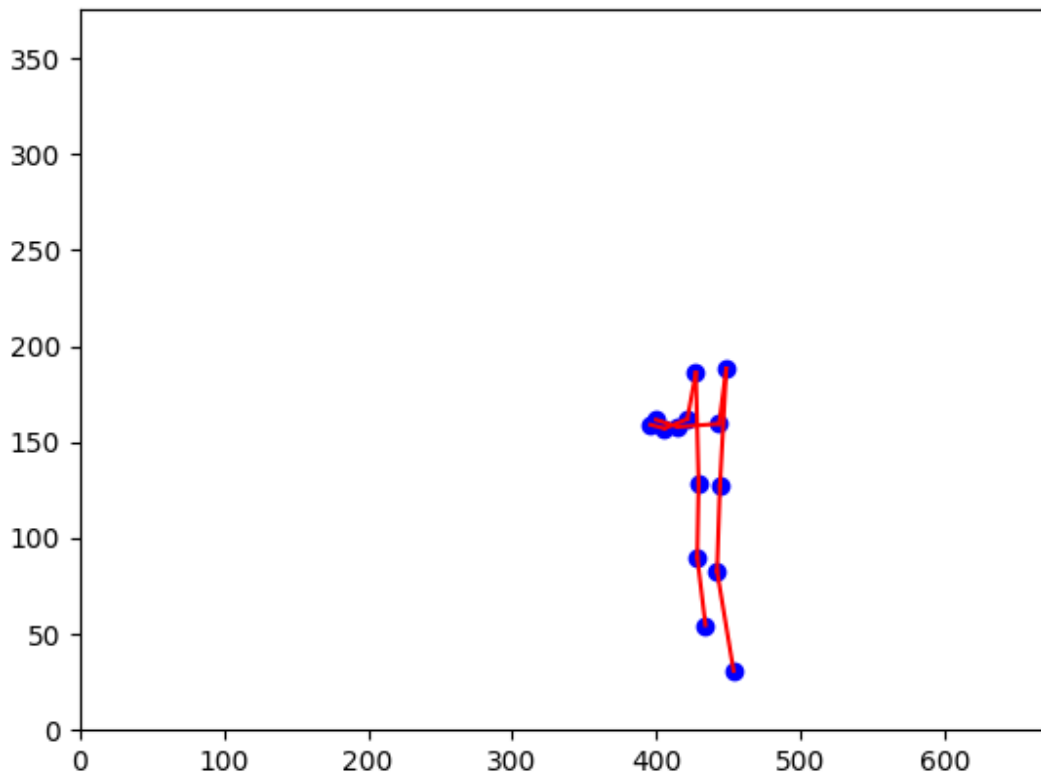


Figure 5.3: Simplified skeleton visualization focusing on key joints relevant to dynamic motion analysis.

Joint Angles and Derivatives Over Time

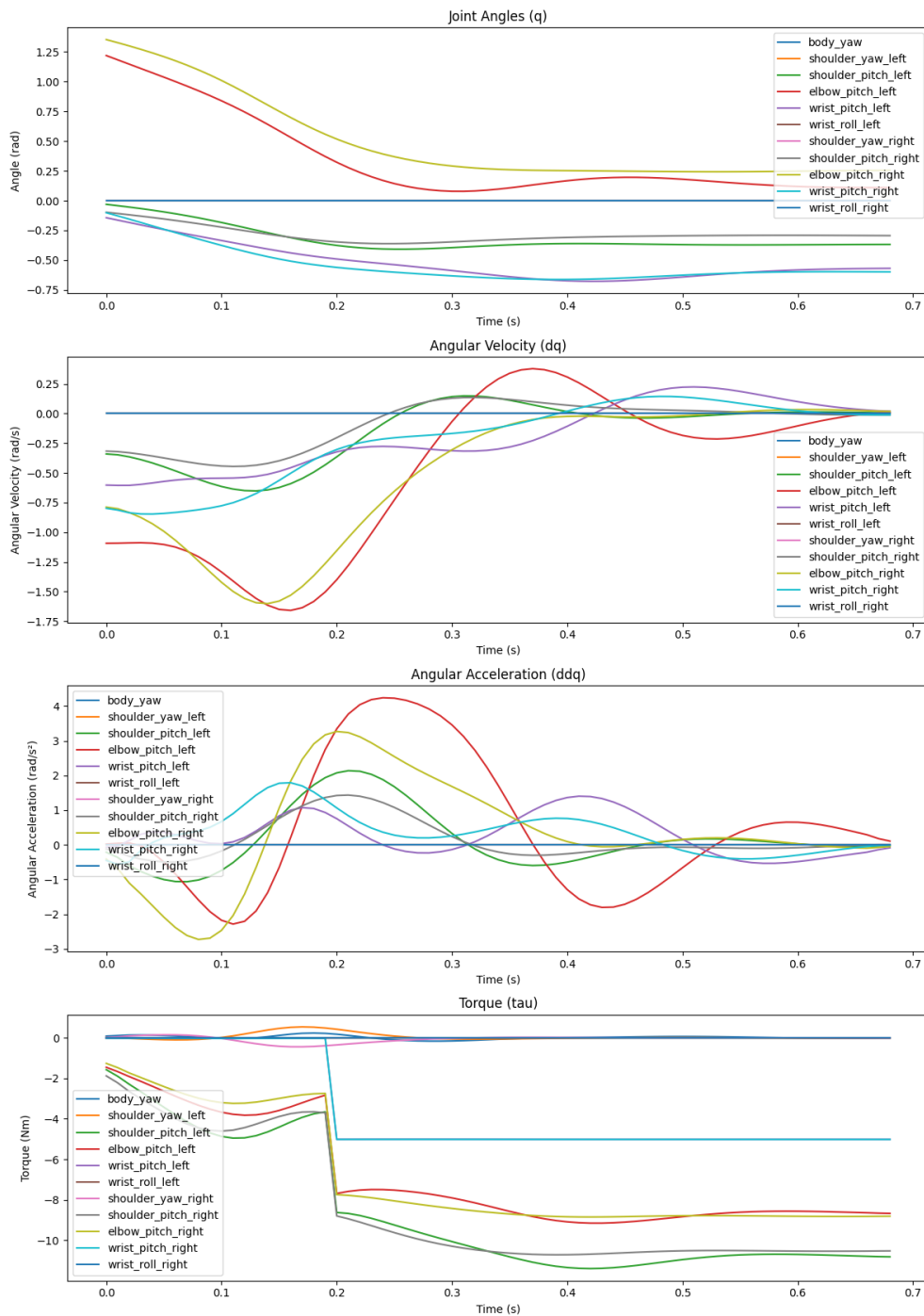


Figure 5.4: Joint Information: plots of joint angles, velocities, accelerations, and torque for the selected joints over time. These dynamics are derived from filtered keypoint trajectories.

5.3 Visuomotor Policy Learning

5.3.1 Problem Setup for Visuomotor Policy Learning

Policy learning from demonstration can be formulated as supervised regression, mapping observations to actions. Given an observation sequence $\mathbf{o} = o_1, \dots, o_t, \dots, o_T$ and corresponding action sequence $\mathbf{a} = a_1, \dots, a_t, \dots, a_T$, where t denotes the time-step, we aim to learn a robust policy $p(\mathbf{a}|\mathbf{o})$ from demonstrations.

5.3.2 Diffusion Policy

The diffusion policy learns this distribution using a conditional diffusion model for actions conditioned on observations. The vanilla diffusion model’s learning objective is:

$$\mathcal{L}(\theta) = \mathbb{E}_{x^0, \epsilon^k} \|\epsilon^k - \epsilon_\theta(x^0 + \epsilon^k, k)\|^2 \quad (5.2)$$

where x^0 is clean data, x^K is noised data, and $\epsilon_\theta(x^k = x^0 + \epsilon^k, k) \approx x^k - x^0$ is a neural network predicting cumulative noise. Superscripts denote denoising steps while subscripts indicate sequence time-steps.

For the action diffusion model, we denoise action sequences conditioned on observations. For each time step t with (o_t, a_t) :

$$\mathcal{L}(\theta) = \mathbb{E}_{a_0, \epsilon^k} \|\epsilon^k - \epsilon_\theta(o_t, a_t^0 + \epsilon^k, k)\|^2 \quad (5.3)$$

where $\epsilon_\theta(o_t, a_t^0 + \epsilon^k, k)$ is a conditional noise prediction network.

Implementation considerations: 1. Video sequence observations require a visual encoder for low-dimensional compression. 2. For temporal consistency, we execute T_a predicted action steps before replanning, using the latest T_o observation steps as input.

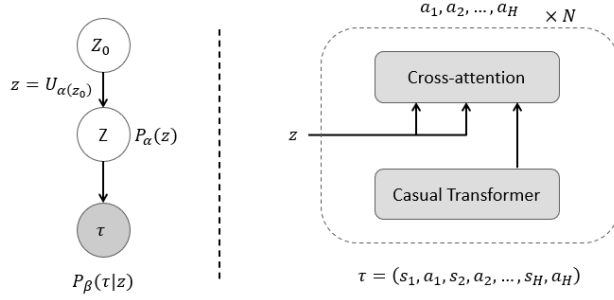


Figure 5.5: Overview of the Latent Plan Transformer (LPT) architecture. The latent plan z is sampled from the prior $p_\alpha(z)$ and conditions the Transformer-based policy generator p_β , which outputs a sequence of actions based on past actions, observations, and the latent plan. This figure illustrates how LPT integrates the latent plan sampling and policy generation processes within a single framework.

5.3.3 Latent Plan Transformer

Latent Plan Transformer (LPT) learns $p(\boldsymbol{\tau})$ ($\boldsymbol{\tau} = (\mathbf{a}, \mathbf{o})$) through factorization:

$$p(\boldsymbol{\tau}) = \int p(\boldsymbol{\tau}, z) dz = \int p_\alpha(z) p_\beta(\boldsymbol{\tau}|z) dz \quad (5.4)$$

where z represents the plan, $p_\alpha(z)$ is the learnable prior model, and $p_\beta(\boldsymbol{\tau}|z)$ is the policy generator factorized autoregressively:

$$p_\beta(\boldsymbol{\tau}|z) = \prod_{t=1}^T p_\beta(\boldsymbol{\tau}_t | \boldsymbol{\tau}_{<t}|z).$$

The implementation uses a Transformer model for p_β , conditioning on observations to predict actions:

$$p_\beta(\boldsymbol{\tau}|z) = \prod_{t=1}^T p(\mathbf{a}_t | \mathbf{a}_{<t}, \mathbf{o}_{<t}, z).$$

For generation, we sample $z \sim p_\alpha(z)$ and generate actions via $p_\beta(\boldsymbol{\tau}|z)$. For replanning with observations \mathbf{o} , we sample $z \sim p(z|\boldsymbol{\tau}) \propto p(z)p(\boldsymbol{\tau}|z)$ initialized from the previous z and then regenerate actions.

As illustrated in Figure 5.5, the LPT framework integrates the latent plan sampling $p_\alpha(z)$ with the Transformer-based policy generator $p_\beta(\tau|z)$ to produce action sequences. This design enables the model to dynamically adapt its action predictions to different contexts and observations, facilitating more robust and flexible behavior in complex tasks.

5.4 Experiments and Results

This section presents preliminary simulation-based evaluations of the proposed method. While real-world hardware experiments have not yet been conducted, the simulation results provide initial insights into the approach’s effectiveness, including comparisons against a baseline diffusion policy.

Simulation Setup

All experiments were conducted in MuJoCo to facilitate controlled testing. RGB-D data was captured at 100 Hz and stored in shared memory for processing. Joint states (positions, velocities, accelerations) were retrieved directly from the simulation. Both non-impact mitigation and impact mitigation motions were tested to examine differences in torque profiles and overall manipulation performance.

Preliminary Timing Results

The following results summarize the current performance in terms of trajectory generation and replanning times. While our model currently demonstrates a higher raw frequency than the baseline diffusion model, these values represent only the direct generation times without any interpolation or further optimization. Future refinements, such as interpolating between actions, may effectively increase the control frequency well beyond the figures presented here.

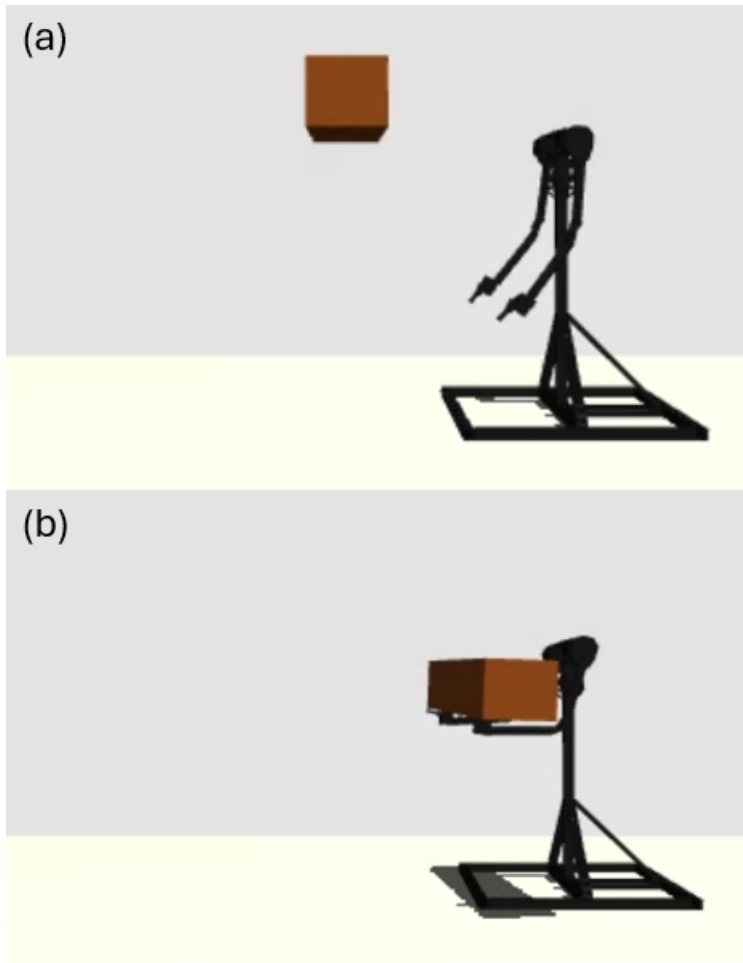


Figure 5.6: Simulation setup illustrating the environment in which the preliminary experiments were conducted.

Key Points:

- Our model achieves a replanning frequency of approximately 18.7 Hz for single-action generation.
- Generating a full 300 action trajectory with our model takes about 6.23 seconds, corresponding to 0.16 Hz.
- The baseline diffusion model produces a 300 action trajectory in about 7.88 seconds, yielding 0.13 Hz.

- These raw frequencies do not yet incorporate techniques like interpolation, which can substantially increase the effective control rate.

Table 5.1: Preliminary Timing Results for Trajectory Generation and Replanning

Model	Task	Time / Frequency
Our Model	Replan + 1 Action	0.0536 s (18.7 Hz)
Our Model (Initial Planning)	300 Actions	6.2274 s (0.16 Hz)
Diffusion Model	300 Actions	7.8756 s (0.13 Hz)

These preliminary findings serve as a baseline for future improvements. With appropriate tuning and interpolation methods, the effective operational frequency is expected to increase, enabling more responsive and efficient control strategies.

5.5 Summary

This chapter introduced the **AVID** framework, which aims to improve dynamic manipulation through the integration of human-inspired strategies, video-derived demonstrations, and explicit consideration of the robot’s dynamic properties. Unlike previous methods that focus solely on motion retargeting, AVID leverages human demonstration data to inform a robot dynamics model, thereby enabling more energy-efficient and robust manipulations. The approach was exemplified through the box-receiving task, a representative dynamic manipulation scenario that balances complexity and tractability, allowing the evaluation of core elements such as timing, contact handling, and continuous replanning.

We detailed a comprehensive data collection and processing pipeline. This included 2D pose estimation, keypoint filtering, skeleton simplification, and inverse dynamics calculations, culminating in joint torque estimates for the target manipulator. The processed data and simplified skeleton served as a foundation for testing different control strategies—ranging

from inverse dynamics control for precise trajectory tracking to impedance and MPC controllers for compliant or unstable trajectories.

Preliminary simulation-based evaluations provided initial insights into AVID’s potential. Although hardware experiments remain a future goal, the simulation results indicated that our model achieves higher raw frequencies for single-action generation and full-trajectory planning compared to a baseline diffusion policy. Despite these early advantages, the current frequencies represent unoptimized conditions without interpolation. The control frequency can be significantly enhanced with appropriate tuning and incremental improvements.

In summary, this work lays the groundwork for more adaptive and capable robotic manipulation in dynamically changing, uncertain environments. By bridging the gap between human demonstrations, robot-specific dynamics, and flexible control frameworks, AVID offers a promising avenue for advancing the field of dynamic manipulation toward safer, more efficient, and more versatile robotic systems.

CHAPTER 6

Conclusion and Future Works

6.1 Conclusion

This dissertation has investigated the synergy among innovative hardware design, control methodologies suited to emerging actuator technologies, advanced motion planning strategies, and machine learning techniques, all aimed at enhancing the capabilities of robotic manipulators in dynamic, human-centric environments. Central to this work is the deployment of proprioceptive actuators, whose inherent backdrivability and torque control capabilities introduce a fundamentally new manipulation paradigm compared to traditional high-gear-ratio actuators. By emphasizing torque control, compliance, and adaptability, these actuators enable more flexible and robust robot-environment interactions. These qualities are becoming increasingly important as we integrate robots into everyday human life.

The low-inertia, proprioceptive dual-arm manipulator developed in this dissertation demonstrates that such systems can excel not only in controlled laboratory settings but also in realistic, application-driven scenarios. We have shown that adopting proprioceptive actuators and focusing on torque optimization strategies (including jerk-minimized trajectories and friction-aware force computations) leads to improved manipulation performance, reduced mechanical stress, and enhanced safety. The successful integration of these principles in tasks as diverse as cooking, box receiving, and friction-informed grasping validates the practical viability of our approaches.

A key insight of this work is the shift away from exclusively position-control-based

methodologies. While position control remains indispensable for certain tasks, the experimental evidence presented here strongly suggests that force and torque-based control schemes provide more efficient, and robust solutions. This dissertation thus serves as a stepping stone toward a new class of manipulators that can actively sense, predict, and adapt to complex physical interactions, paving the way for future robots that operate safely and reliably in uncertain, dynamically changing conditions.

6.2 Future Plans

Although the advancements reported here mark an important milestone, several avenues for future research can further extend the capabilities and applicability of our approaches:

1. Enhanced Torque-Optimized Control Frameworks

While we have demonstrated successful applications of inverse dynamics and impedance controllers, future efforts should focus on developing unified, torque-centric control frameworks capable of seamlessly blending these strategies. Such controllers would better handle sudden disturbances, high-impact interactions, and the delicate balance between compliance and stability.

2. Advanced Trajectory Planning and Real-Time Adaptation

The incorporation of minimum-jerk trajectory optimization highlights the promise of smoother, more natural manipulations. Future research should explore even more sophisticated planning algorithms that can incorporate real-time sensory feedback, collision avoidance, and dynamic adjustments in response to changing goals or constraints, all while maintaining computational efficiency suitable for real-world deployments.

3. Broader Friction Data and Material Modeling

The friction-aware strategies introduced here rely on accurate friction estimates derived from specialized datasets. Enlarging the scope of available friction data, covering a wider range of materials, textures, and environmental conditions, will allow robotic systems to generalize more effectively. Enhanced friction modeling, combined with predictive learning, can lead to even more reliable and adaptive manipulation in unstructured environments.

4. Refinement and Expansion of Video Imitation Learning with Contact Dynamics

While initial results in translating human demonstrations into robot joint trajectories are encouraging, the immediate priority is to refine and optimize the current methodology through meticulous parameter tuning. By first ensuring stable and high-performance imitation policies, we establish a robust foundation for future enhancements. Once these performance benchmarks are met, the next step involves integrating comprehensive contact and dynamics modeling into the learning framework. This expanded approach will allow the robot not only to replicate human-like motions but also to internalize the underlying physical principles governing those motions. We can significantly enhance the robustness, transferability, and overall realism of video-driven imitation learning through a two-stage process. First, we will solidify the current approach, particularly focusing on hardware testing. Next, we will enrich this framework by incorporating contact dynamics and more detailed datasets. This will help bridge the gap between simulated training and real-world execution.

5. Extension to Mobile Manipulators and Complex Real-World Tasks

Building upon the insights gained thus far, an immediate goal is to extend these approaches to mobile manipulator platforms. By integrating proprioceptive actuators, torque-based control, and learning-driven adaptation with a mobile base, we can evaluate a broader range

of manipulation tasks in dynamic and unstructured environments. This will enable us to test how well the methods generalize to mobile robots performing diverse and challenging operations, including those that require precise coordination and real-time adaptation.

Ultimately, the aim is to tackle complex real-world problems that demand both mobility and dexterity. For instance, tasks like using a hammer to drive in nails (Figure 6.1) or painting surfaces (Figure 6.2) exemplify the kind of intricate, force-sensitive operations robots should master. By extending our current frameworks to mobile platforms and validating them through tasks such as hammering and painting, we take a significant step toward deploying robots that can seamlessly integrate into everyday human activities, contributing to areas such as construction, maintenance, and creative craftsmanship.

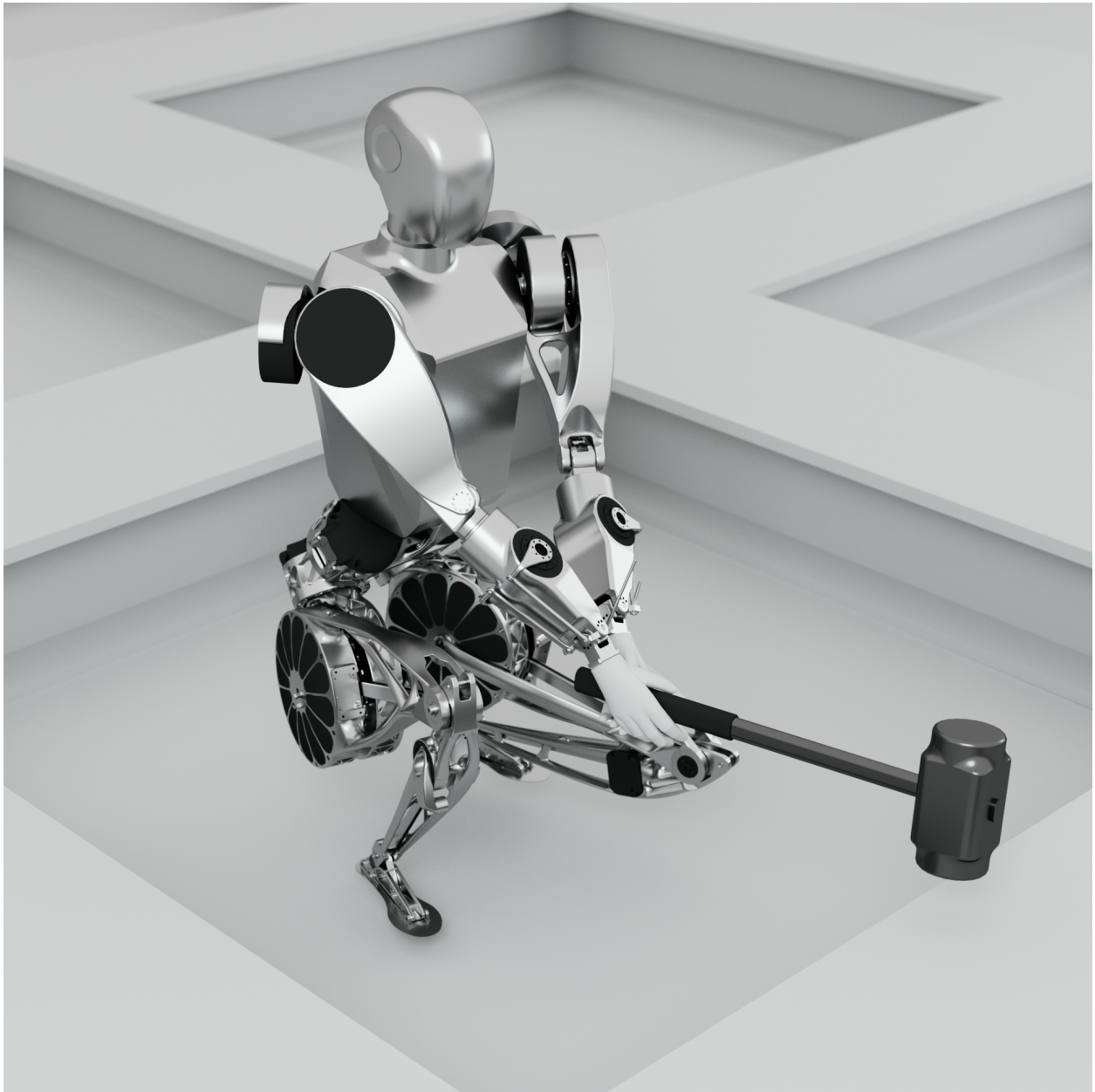


Figure 6.1: Impact and force-based tasks, such as hammering, represent challenging real-world scenarios for mobile manipulators.



Figure 6.2: Painting tasks, requiring careful force modulation and dexterity, illustrate the potential for mobile manipulators in complex human-centric environments.

REFERENCES

- [Ahn23] Min Sung Ahn. *Development and Real-Time Optimization-based Control of a Full-sized Humanoid for Dynamic Walking and Running*. University of California, Los Angeles, 2023.
- [BDG85] James E Bobrow, Steven Dubowsky, and John S Gibson. “Time-optimal control of robotic manipulators along specified paths.” *The International Journal of Robotics Research*, **4**(3):3–17, 1985.
- [BMA14] Jeannette Bohg, Aitor Morales, Tamim Asfour, and Danica Kragic. “Data-driven grasp synthesis—A survey.” *IEEE Transactions on Robotics*, **30**(2):289–309, 2014.
- [CC00] Daniela Constantinescu and Elizabeth A Croft. “Smooth and time-optimal trajectory planning for industrial manipulators along specified paths.” *Journal of robotic systems*, **17**(5):233–249, 2000.
- [CJP11] Peter I Corke, Witold Jachimczyk, and Remo Pillat. *Robotics, vision and control: fundamental algorithms in MATLAB*, volume 73. Springer, 2011.
- [CLY21] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. “Transunet: Transformers make strong encoders for medical image segmentation.” *arXiv preprint arXiv:2102.04306*, 2021.
- [Con20] MMPose Contributors. “OpenMMLab Pose Estimation Toolbox and Benchmark.” <https://github.com/open-mmlab/mmpose>, 2020.
- [Cra05] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson/Prentice Hall, 3rd edition, 2005.
- [CVG14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.” In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.
- [DMV09] Ravinder S Dahiya, Giorgio Metta, Maurizio Valle, and Giulio Sandini. “Tactile sensing—from humans to humanoids.” *IEEE transactions on robotics*, **26**(1):1–20, 2009.
- [DV88] Joris De Schutter and Hendrik Van Brussel. “Compliant Robot Motion I: A Formalism for Specifying Compliant Motion Tasks.” *The International Journal of Robotics Research*, **7**(4):3–17, 1988.

- [EML19] Shirine El Zaatari, Mohamed Marei, Weidong Li, and Zahid Usman. “Cobot programming for collaborative industrial tasks: An overview.” *Robotics and Autonomous Systems*, **116**:162–180, 2019.
- [FH85a] Tamar Flash and Neville Hogan. “Coordination of arm movements: an experimentally confirmed mathematical model.” *Journal of Neuroscience*, **5**(7):1688–1703, 1985.
- [FH85b] Tamar Flash and Neville Hogan. “The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model.” *Journal of Neuroscience*, **5**(7):1688–1703, 1985.
- [GMY19] David V Gealy, Stephen McKinley, Brent Yi, Philipp Wu, Phillip R Downey, Greg Balke, Allan Zhao, Menglong Guo, Rachel Thomasson, Anthony Sinclair, et al. “Quasi-direct drive for low-cost compliant robotic manipulation.” In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 437–443. IEEE, 2019.
- [GV18] Ambarish Goswami and Prahlad Vadakkepat. *Humanoid robotics: a reference*. Springer Publishing Company, Incorporated, 2018.
- [HAH08] Sami Haddadin, Alin Albu-Schäffer, and Gerd Hirzinger. “The role of the robot mass and velocity in physical human-robot interaction—Part I: Non-constrained blunt impacts.” In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1331–1338. IEEE, 2008.
- [HCW23] Binghao Huang, Yuanpei Chen, Tianyu Wang, Yuzhe Qin, Yaodong Yang, Nikolay Atanasov, and Xiaolong Wang. “Dynamic handover: Throw and catch with bimanual hands.” *arXiv preprint arXiv:2309.05655*, 2023.
- [HGE17] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. “Imitation learning: A survey of learning methods.” *ACM Computing Surveys (CSUR)*, **50**(2):1–35, 2017.
- [HKS10] Brad Hamner, Seth Koterba, Jane Shi, Reid Simmons, and Sanjiv Singh. “An autonomous mobile manipulator for assembly tasks.” *Autonomous Robots*, **28**:131–149, 2010.
- [HO07] Masato Hirose and Kenichi Ogawa. “Honda humanoid robots development.” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **365**(1850):11–19, 2007.
- [Ho22] Daniel Ho et al. “Diffusion policy: Visuomotor policy learning via action diffusion.” *arXiv preprint arXiv:2205.12524*, 2022.

- [Hog85] Neville Hogan. “Impedance Control: An Approach to Manipulation.” *Journal of Dynamic Systems, Measurement, and Control*, **107**(1):1–24, 1985.
- [HS97] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory.” *Neural Computation*, **9**(8):1735–1780, 1997.
- [HS05] Won Hong and Jean-Jacques E Slotine. “Experiments in hand-eye coordination using active vision.” In *Experimental Robotics IV: The 4th International Symposium, Stanford, California, June 30–July 2, 1995*, pp. 130–139. Springer, 2005.
- [KAS16] Mina Kamel, Kostas Alexis, and Roland Siegwart. “Design and modeling of dexterous aerial manipulator.” In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4870–4876. IEEE, 2016.
- [KC02] Danica Kragic, Henrik I Christensen, et al. “Survey on visual servoing for manipulation.” *Computational Vision and Active Perception Laboratory, Fiskartorpsv*, **15**:2002, 2002.
- [KCK13] Suseong Kim, Seungwon Choi, and H Jin Kim. “Aerial manipulation using a quadrotor with a two dof robotic arm.” In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4990–4995. IEEE, 2013.
- [Kha87] Oussama Khatib. “A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation.” *IEEE Journal of Robotics and Automation*, **3**(1):43–53, 1987.
- [Kim15] Yong-Jae Kim. “Design of low inertia manipulator with high stiffness and strength using tension amplifying mechanisms.” In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5850–5856. IEEE, 2015.
- [Kim17] Yong-Jae Kim. “Anthropomorphic low-inertia high-stiffness manipulator for high-speed safe interaction.” *IEEE Transactions on robotics*, **33**(6):1358–1374, 2017.
- [KK23] Hamidreza Kasaei and Mohammadreza Kasaei. “Throwing objects into a moving basket while avoiding obstacles.” In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3051–3057. IEEE, 2023.
- [KSP04] Oussama Khatib, Luis Sentis, Jaeheung Park, and John Warren. “Whole-Body Dynamic Behavior and Control of Human-Like Robots.” *International Journal of Humanoid Robotics*, **1**(1):29–43, 2004.
- [KXZ24] Deqian Kong, Dehong Xu, Minglu Zhao, Bo Pang, Jianwen Xie, Andrew Lizarraga, Yuhao Huang, Sirui Xie, and Ying Nian Wu. “Latent Plan Transformer for Trajectory Abstraction: Planning as Latent Space Inference.” In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

- [LC23] Shihan Lu and Heather Culbertson. “Active Acoustic Sensing for Robot Manipulation.” In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3161–3168. IEEE, 2023.
- [LFD16] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. “End-to-end training of deep visuomotor policies.” *Journal of Machine Learning Research*, **17**(1):1334–1373, 2016.
- [LP17] Kevin M Lynch and Frank C Park. *Modern robotics*. Cambridge University Press, 2017.
- [LPK18] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection.” *International Journal of Robotics Research*, **37**(4-5):421–436, 2018.
- [LSB23] Jaemin Lee, Mingyo Seo, Andrew Bylard, Robert Sun, and Luis Sentis. “Real-time model predictive control for industrial manipulators with singularity-tolerant hierarchical task control.” In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12282–12288. IEEE, 2023.
- [LWZ23] Fengbo Lan, Shengjie Wang, Yunzhe Zhang, Haotian Xu, Oluwatosin Oseni, Yang Gao, and Tao Zhang. “DexCatch: Learning to Catch Arbitrary Objects with Dexterous Hands.” *arXiv preprint arXiv:2310.08809*, 2023.
- [Mac02] Jan M. Maciejowski. *Predictive Control with Constraints*. Pearson Education, 2002.
- [Mas81] Matthew T. Mason. “Compliance and Force Control for Computer Controlled Manipulators.” *IEEE Transactions on Systems, Man, and Cybernetics*, **11**(6):418–432, 1981.
- [MBS20] Nicolas Mansard, Baptiste Bottemanne, and Guilhem Saurel. “Integrated Motion Generation and Control: The Whole-Body Paradigm.” *Annual Review of Control, Robotics, and Autonomous Systems*, **3**(1):1–33, 2020.
- [MC12] Manfred Morari and Eduardo F. Camacho. *Model Predictive Control*. Springer, 2012.
- [MK11] Daniel Mellinger and Vijay Kumar. “Minimum snap trajectory generation and control for quadrotors.” In *2011 IEEE international conference on robotics and automation*, pp. 2520–2525. IEEE, 2011.
- [MRR00] David Q. Mayne, James B. Rawlings, Carmen V. Rao, and Peter O. M. Sokaert. “Constrained Model Predictive Control: Stability and Optimality.” *Automatica*, **36**(6):789–814, 2000.

- [NNA21] Donghun Noh, Hyunwoo Nam, Min Sung Ahn, Hosik Chae, Sangjoon Lee, Kyle Gillespie, and Dennis Hong. “Surface material dataset for robotics applications (smdra): A dataset with friction coefficient and rgb-d for surface segmentation.” In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 6275–6281. IEEE, 2021.
- [OAS06] Yu Ogura, Hiroyuki Aikawa, Kazushi Shimomura, Hideki Kondo, Akitoshi Morishima, Hun-ok Lim, and Atsuo Takanishi. “Development of a new humanoid robot WABIAN-2.” In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 76–81. IEEE, 2006.
- [PG12] YD Patel and PM George. “Parallel manipulators applications—a survey.” *Modern Mechanical Engineering*, **2**(03):57–64, 2012.
- [Pha14] Quang-Cuong Pham. “A general, fast, and robust implementation of the time-optimal path parameterization algorithm.” *IEEE Transactions on Robotics*, **30**(6):1533–1540, 2014.
- [PP18] Hung Pham and Quang-Cuong Pham. “A new approach to time-optimal path parameterization based on reachability analysis.” *IEEE Transactions on Robotics*, **34**(3):645–659, 2018.
- [RBM13] Ludovic Righetti, Jonas Buchli, Mitsuhiro Mistry, and Stefan Schaal. “Inverse Dynamics Control of Floating-Base Robots with External Constraints: A Unified Framework for Whole-Body Control.” In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1085–1090, 2013.
- [RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation.” In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.
- [RLO18] Fabio Ruggiero, Vincenzo Lippiello, and Anibal Ollero. “Aerial manipulation: A literature review.” *IEEE Robotics and Automation Letters*, **3**(3):1957–1964, 2018.
- [SHV06] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, 2006.
- [SK16] Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer, 2nd edition, 2016.
- [SKY18] Hansol Song, Yun-Soo Kim, Junsuk Yoon, Seong-Ho Yun, Jiwon Seo, and Yong-Jae Kim. “Development of low-inertia high-stiffness manipulator LIMS2 for high-speed manipulation of foldable objects.” In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4145–4151. IEEE, 2018.

- [SSS10] Dragoljub Surdilovic, Gerhard Schreck, and Uwe Schmidt. “Development of collaborative robots (cobots) for flexible human-integrated assembly automation.” In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pp. 1–8. VDE, 2010.
- [SSV09] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer, 2009.
- [SWO12] Sangok Seok, Albert Wang, David Otten, and Sangbae Kim. “Actuator design for high force proprioceptive control in fast legged locomotion.” In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1970–1975. IEEE, 2012.
- [THK16] James Trevelyan, William R Hamel, and Sung-Chul Kang. “Robotics in hazardous applications.” *Springer handbook of robotics*, pp. 1521–1548, 2016.
- [TSA23] Shantanu Thakar, Srivatsan Srinivasan, Sarah Al-Hussaini, Prahar M Bhatt, Pradeep Rajendran, Yeo Jung Yoon, Neel Dhanaraj, Rishi K Malhan, Matthias Schmid, Venkat N Krovi, et al. “A survey of wheeled mobile manipulation: A decision-making perspective.” *Journal of Mechanisms and Robotics*, **15**(2):020801, 2023.
- [WWS17] Patrick M Wensing, Albert Wang, Sangok Seok, David Otten, Jeffrey Lang, and Sangbae Kim. “Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots.” *Ieee transactions on robotics*, **33**(3):509–522, 2017.
- [YGH12] Mehrdad Yazdani, Geoffrey Gamble, Gavin Henderson, and Robert Hecht-Nielsen. “A simple control policy for achieving minimum jerk trajectories.” *Neural Networks*, **27**:74–80, 2012.
- [YZO17] Wenzhen Yuan, Chenzhuo Zhu, Andrew Owens, Mandayam A Srinivasan, and Edward H Adelson. “Shape-independent hardness estimation using deep learning and a gelsight tactile sensor.” In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 951–958. IEEE, 2017.
- [ZHL20] Ali Zahavi, Shahriar Najafi Haeri, Dhanushka Chamara Liyanage, and Mart Tamre. “A dual-arm robot for collaborative vision-based object classification.” In *2020 17th Biennial Baltic Electronics Conference (BEC)*, pp. 1–5. IEEE, 2020.
- [ZLT23] Jianzhuang Zhao, Gustavo JG Lahr, Francesco Tassi, Alessandro Santopaolo, Elena De Momi, and Arash Ajoudani. “Impact-friendly object catching at non-zero velocity based on combined optimization and learning.” In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4428–4435. IEEE, 2023.

- [ZSL20] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. “Tossingbot: Learning to throw arbitrary objects with residual physics.” *IEEE Transactions on Robotics*, **36**(4):1307–1319, 2020.