Title
Physical Human-Robot Interaction with Exoskeletons

Author
Sun, Jianwei

Publication Date
2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Physical Human-Robot Interaction with Exoskeletons

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mechanical Engineering

by

Jianwei Sun

2024

ABSTRACT OF THE DISSERTATION

Physical Human-Robot Interaction with Exoskeletons

by

Jianwei Sun

Doctor of Philosophy in Mechanical Engineering

University of California, Los Angeles, 2024

Professor Jacob Rosen, Chair

Physical interaction between humans and robots is becoming more widespread in society due to major research efforts in developing intelligent robotics in numerous industries: health care, manufacturing, transportation, energy, etc. The coupled human-robot system allows the strengths of one part to overcome the limitations of the other; the high-level task planning and problem-solving abilities of the human are complemented by the accuracy, strength, reliability, and repeatability of the robot.

One technology to benefit from these advancements is the exoskeleton, which has applications primarily in robot-assisted rehabilitation and human augmentation. Admittance control enables motion of the exoskeleton by generating reference trajectories corresponding to a lightweight virtual system that responds to human-applied forces. The exoskeleton's controller then tracks the reference, giving the illusion that the exoskeleton's dynamics are those of the lightweight system. The ease at which the exoskeleton moves with the operator is known as transparency. The virtual system can emulate many desired dynamics, such as those of point masses or rigid bodies. This flexibility is foundational for more advanced applications, such as having the exoskeleton also provide assistive forces during rehabilitation or restricting motions to be within a safe workspace.

Naturally, when coupling a human to an exoskeleton, there are important safety, reliability, and performance considerations. Safety is often addressed at all levels of the system: mechanics, controller, and reference generation, and is further explored in this work. The feedback connection between the human and the exoskeleton also creates possible stability issues, which may not exist in either system independently. Delay-induced instability and a mechanism for mitigation are also explored.

Furthermore, improving transparency can conflict with wearability. However, for robot-assisted rehabilitation, such a trade-off may be necessary. Thus, achieving comparable behavior for a system that has a simpler human-robot interface is of importance and is also explored.

Finally, the flexibility of virtual dynamics can enable behavior that is not otherwise possible. For instance, creating virtually constrained motion for path-guided rehabilitation and virtual reality-based object manipulation can greatly improve robot-assisted rehabilitation. These examples showcase the significant potential of robotics for the field of rehabilitation.

The dissertation of Jianwei Sun is approved.

Veronica Santos

Tetsuya Iwasaki

Jason L. Speyer

Jacob Rosen, Committee Chair

University of California, Los Angeles

2024

To my parents.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Firstly, I would like to express my deepest gratitude to my research advisor, Professor Jacob Rosen. You welcomed me into the Bionics Lab as a visiting researcher, and have always encouraged me and taught me to become a better researcher. I am grateful for the opportunity you have given me. I wish to also thank my Ph.D. committee: Professors Veronica Santos, Tetsuya Iwasaki, Jason Speyer, and Jacob Rosen. Your guidance over the course of my Ph.D. has been invaluable, and I am incredibly fortunate for your support. Special thanks to Professor Paulo Tabuada; our discussions during your nonlinear and geometric control classes and receiving your guidance about which courses to take have helped me greatly. Professor Khalid Jawed, thank you for the opportunity to work with you for the MATLAB courses. I sincerely appreciate everything you've done for me, as well as your trust in me, even though it was my first time TA'ing. Dr. Ji Ma, thank you for all your help and guidance throughout my Ph.D. You are one of the most intelligent people from whom I had the opportunity to learn, and I am grateful for your part in my Ph.D.

I would also like to thank all of my friends at the Bionics Lab. It has been an incredible journey, and I could not have done it without your support. Yang, my former supervisor and first contact at UCLA, your involvement means so much. You showed me how to research and write academic papers, and I am fortunate to have been able to work together. Hao, your high-spirited and light-hearted personality always made the Bionics Lab feel lively. I am grateful for all the fun we shared, including the Thanksgiving meals (which was how I learned of my cat allergy), go-karting, our endless discussions about coding, and even teaching me how to manipulate the UCLA swipe market. Changyeob, I am thankful for everything I learned from you. It has truly been a blast; from eating countless lunches and dinners together, to going to my first Basketball game, I'm glad you were a part of my time here. Peter, your friendship and collaboration mean so much to me. I enjoyed every aspect of working together on our papers and just chatting about anything. You have always been open to talk with and I am fortunate for our time together. Erik, from working on the V-Rex, to our mini road-trip to SCR, it has truly been a pleasure. I appreciate all our conversations, which ranged anywhere from academics to programming, from random facts about space to riddles, and from Photoshop tips to even the order of meat to cook in KBBQ. Thank you. Sahba, even though our overlap was small, I still learned much from you; you are a great role model. Seungmin, thank you for being a part of my Ph.D. It was always great to talk with you, and I enjoyed working together on our C247 project - also thank you for helping me with my car! Haoran, I remember the Kawasaki feasibility study that you, Erik, and I conducted; it was an unforgettable experience working late into the night but finally getting it to work. Yasamin, I enjoyed our time together, whether that's working on a paper, TA'ing a course, playing an online game, or even trying out a martial art. I value our friendship and I'm glad you were a big part of my time here. Mianzhi, thank you for all the programming and controls discussions and fun we've had over the years. It has always a pleasure to chat and game with you. Hanbyeol, Bita, Matt, Rick, Xiaochen, and Guanyu - I am cheering for you! Thank you for being a part of my journey. Andrew, you are brilliant and I am glad we got to work together; I especially enjoyed all our late night discussions about math.

Rocky, being one of the first people I met when I started this degree, you have always been friendly and easy going. I always enjoyed our conversations, be it academics or life. I am lucky for our time at UCLA, and I look forward to trying out the next online game together. Lisa, when we first met in Singapore and I discovered your humor, I knew we'd get along. From our countless Facebook messages to all the memes we've exchanged over the years, your positiveness has always encouraged me. Thank you for your friendship.

Wenyi, my partner and best friend, I cannot begin to express my appreciation for all the happiness you have brought into my life. I cherish all the moments we have spent together, and I am excited for our future. You have always been supportive, and your encouragement has taught me to be optimistic and level-headed. I am so lucky you are in my life.

To my parents, words alone cannot describe your support and love for me. You taught me how to be hard-working and determined, and I will carry these life lessons with me forever. I love you.

VITA

2009–2012	International Baccalaureate Diploma, Western Canada High School
2012–2017	B.A.Sc. in Engineering Science (Electrical and Computer Engineering), University of Toronto
2014	Researcher, Mechanical Engineering, National University of Singapore
2015–2016	Hardware Intern, Wearable Systems, Apple
2016	Researcher, Institute for Dynamic Systems and Control, ETH Zürich
2016–2017	Researcher, Reconfigurable Antenna Laboratory, University of Toronto
2017	Software Intern, Programmable Solutions Group, Intel
2017–2019	M.Sc. in Electrical Engineering and Information Technology, ETH Zürich
2017–2018	Researcher, Computer Engineering and Networks Laboratory, ETH Zürich
2018	Researcher, Institute for Dynamic Systems and Control, ETH Zürich
2018–2019	Hardware Intern, Wearable Systems, Apple
2019–2024	Researcher, Bionics Lab, UCLA
2019–2024	Teaching Assistant, UCLA
2022	Software Intern, Autonomy, Skydio

PUBLICATIONS

J. Sun, Y. Foroutani, J. Rosen, "Virtually Constrained Admittance Control using Feedback Linearization for Physical Human-Robot Interaction with Rehabilitation Exoskeletons." Under review. [1]

J. Sun, E. H. Kramer, J. Rosen, "A Safety-Focused Admittance Control Approach for Physical Human-Robot Interaction with Rigid Multi-Arm Serial Link Exoskeletons." Under review. [2]

P. W. Ferguson, **J. Sun** and J. Rosen, "On the OTHER Hand: A Bimanual, Reconfigurable Hand Exoskeleton with Opposable Thumbs for Use with Upper Limb Exoskeletons." Under review. [3]

J. Sun, P. W. Ferguson and J. Rosen, "Suppressing Delay-Induced Oscillations in Physical Human-Robot Interaction with an Upper-Limb Exoskeleton using Rate-Limiting," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, 2022, pp. 6695-6701, doi: 10.1109/IROS47612.2022.9981943. [4]

J. Sun, Y. Shen, J. Rosen, "Sensor Reduction, Estimation, and Control of an Upper-Limb Exoskeleton," in *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 1012-1019, April 2021, doi: 10.1109/LRA.2021.3056366. [5]

Y. Shen, **J. Sun**, J. Ma, J. Rosen, "Admittance Control Scheme Comparison of EXO-UL8: A Dual-Arm Exoskeleton Robotic System," *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, Toronto, ON, Canada, 2019, pp. 611-617, doi: 10.1109/ICORR.2019.8779545. [6]

J. Sun and R. Gill, "Vehicle Platoon Control with Virtual Path Constraints," *2019 IEEE Conference on Control Technology and Applications (CCTA)*, Hong Kong, China, 2019, pp. 456-461, doi: 10.1109/CCTA.2019.8920555. [7]

C. Geaney, **J. Sun**, S. V. Hum, E. Martinez-de-Rioja, and J. A. Encinar, "Synthesis of a multi-beam dual reflectarray antenna using genetic algorithms," *2017 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting (AP-S/URSI)*, San Diego, CA, USA, 2017, pp. 1179-1180, doi: 10.1109/APUSNCURSINRSM.2017.8072632. [8]

CHAPTER 1

Introduction

The term pHRI broadly describes any system in which a human interacts with a robotic device in order to accomplish some task. Examples of pHRI include rehabilitation robots for physical therapy [9–16], teleoperation surgical robots for conducting surgeries [17, 18], exo-suit robots for strength augmentation [19–21], collaborative robots [22–24], and so on. In all these examples, the robot and human are combined together so that the strengths of one overlap with the limitations of the other. Humans are intelligent, capable of high level task planning, and able to adapt to complex environments. On the other hand, robots are robust, strong, accurate, repeatable, can operate in harsh or uninhabitable environments, and can record data from numerous sensors. Complementing the two results in a system in which the human plans high level tasks and the robot executes those tasks, creating an efficient system of combined strengths.

Within pHRI, the robot's autonomy can classify the interaction into one of three levels [25] as shown in Fig. 1.1.

1. At level 1, the human has complete control over the robot's motions. In this scenario, the robot serves more as an extension of the human, rather than another agent. Typical examples of this level of cooperation are seen in teleoperation [26, 27] or high-transparency exoskeleton control [5, 28, 29].

2. At level 2, control of the robot is shared between the robot and the human. The human actively controls parts of the robot's movement, while autonomous controllers in the robot handle other aspects of the motion. Human-robot cooperation at this

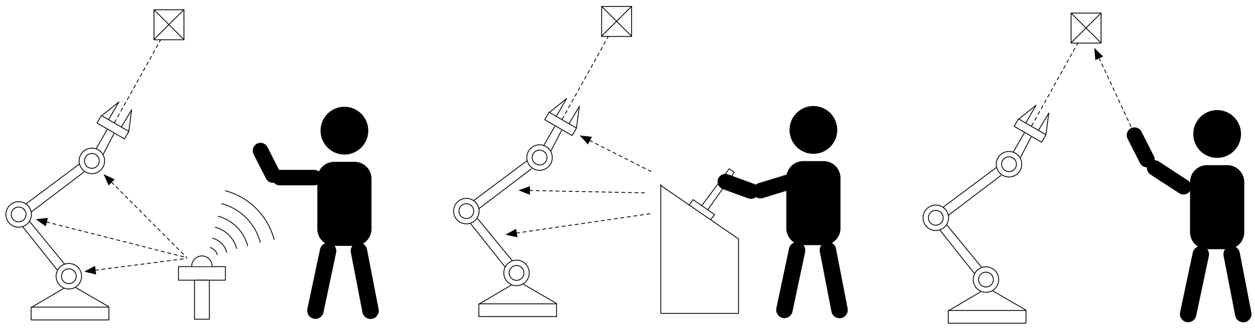


Figure 1.1: One way to classify pHRI is by the robot's autonomy, which can be one of three levels: 1) the human completely controls the robot, 2) the human and robot share control, and 3) the human only specifies only the end-objective.

level can be designed such that the human handles the intelligent high-level aspects, such as path-planning, while the robot handles the other aspects, such as collision avoidance [30].

3. At level 3, the human specifies only the end-objective, and the robot takes care of the other aspects of the motion: path planning, collision avoidance, object handling, etc.

Using the classification described above, exoskeleton devices fall under level 1. Other level 1 interactions can include teleoperation [26, 27], in which a human's input is digitized, transmitted over a network, and reconstructed by a robotic device. For example, teleoperation can be used in dangerous environments, where a human cannot physically be present, or for robot-assisted surgery, where it may be more time-efficient for the surgeon to avoid traveling to the location of each operation [26].

In level 2 interactions, the robot has more autonomy and can perform more complex tasks. The robot is no longer a physical extension of the human operator, but rather it can complement the human in performing the task at hand. Examples of such systems include industrial manipulators with automatic collision avoidance [31], human-controlled robotic platoons [7], or intelligent prostheses that can infer the task that the human is trying to perform [32]. Some literature also refers to robots in this category as “cobots” - a portmanteau of “collaborative” and ”robots” [33].

In level 3, the robot is almost completely autonomous and the human only provides high-level objectives. Autopilot in autonomous cars [34], autonomous cooking robots [35], warehouse organization robots [36], etc, are examples of systems in this category. The physical human-robot interaction at this level is limited, so this dissertation primarily focuses on level 1 and 2 interactions.

1.1 Admittance Control

The aspect of pHRI concerned with moving a robot according to a human operator’s input, such as force, has motivated the use of admittance control. This methodology is one of four input to output combinations from the robotics literature: admittance control (force to position), impedance control (position to force), force control (force to force), and position control (position to position) [37–40]. In this methodology, the controller generates reference signals from human-applied forces, based on the dynamics of some virtual model. The internal controllers of the robot then move it according to these reference signals, so that from the perspective of the operator, the robot appears to move like the virtual model. By designing the model to represent a lightweight system, the robot appears to move effortlessly to human-applied input. The term *transparency* is typically used to describe how well the robot’s motions follow the human’s intentions.

Since the model generating the robot’s motions is virtual, there is freedom in designing the model to capture the desired behavior. For instance, the model can be virtually constrained so that its trajectories never leave certain safety bounding regions. The model can also emulate dynamics with entirely different parameters, e.g., a small point mass to enable fast motions. The flexibility in designing the virtual system gives it utility; however, the farther the virtual dynamics deviate from the physical dynamics, the more difficult it is for the robot’s controller to track the virtual state. Thus, achieving transparent behavior is often more complicated than simply picking a virtual system with minimum inertia and damping. The topic of admittance control and its implementation is explored in Chapter 3.

1.2 Safety

Safety is paramount for any pHRI system. This is especially the case for systems in which the operator is attached to the robot, such as with exoskeletons, or in applications in which the operator has reduced motion or strength, such as in robot-assisted stroke rehabilitation. The importance of safety has motivated numerous approaches that try to address safety at various levels of the software and hardware stack. Electrical and mechanical fail-safes, such as joint limits, power e-stops, breakaway attachments, etc, are often simple to implement and provide basic safety. However, due to their simplicity, they often cannot anticipate dangerous situations, and must instead either react to the situation (e.g., joint limits) or rely on external input (e.g., requiring another operator to press an e-stop button). These detract from their reliability and force them to become last-resort measures. Thus, safety is often also addressed at the controller level. In these cases, more sophisticated measures of safety can be implemented, some of which can even predict when an unsafe situation may occur and preemptively pause the robotic system.

Controller-level safety can monitor important signals within the robot, and automatically respond to situations in which any of these signals exceed predefined limits. For instance, monitoring velocity can ensure that the robot is stopped before its velocity exceeds a value at which potential collisions become dangerous. Monitoring currents in the motors can detect when external forces act on the robot, such as during collisions. Situations such like these cannot simply be prevented from only mechanical joint limits, so they must be handled at the controller level. Implementing safety at the controller level is also beneficial in that these methods not only check for dangerous situations, but can also automatically stop the robot when necessary without relying on external human input.

In addition to reaction-based safety methods, there are also anticipatory approaches. For instance, reference-level approaches can generate trajectories that avoid kinematic singularities or joint limits. These approaches can be very effective since they avoid dangerous situations in the first place. Reference-level approaches are further explored in Chapter 4.

Even when a robotic system is well-designed and incorporates safety throughout, incorporating it in a pHRI setting can cause new dangerous situations for which the robot was not designed. For such HITL systems, the robot's motion simultaneously reacts to and influences that of the human, which can induce phenomena not present in the robot's dynamics. For instance, the time-delay for muscle contraction signals in a human may not pose a great issue in *activities of daily living* (ADLs), but in a exoskeleton with stiff virtual dynamics, they can cause instability of the closed-loop HITL system. The coupled system may have additional dynamics for which neither the robot designer nor the human can anticipate and compensate. Delay-induced instability is one such possibility, and is explored in Chapter 5. In general, the feedback connection of stable systems is not necessarily stable, so additional safety measures must be implemented.

1.3 Sensor Fusion

Sensing and estimation in pHRI includes estimating human intent, which focuses on identifying what the operator is currently doing or plans to do. Knowledge of the intent can then be used as inputs to admittance controllers or other interaction controllers. In the literature, various techniques have been explored, including *surface electromyography* (sEMG)-based techniques [41–45], *electroencephalogram* (EEG)-based "brain-machine interfaces" [46–48], vision-based techniques [49–51], and force measurement and estimation methods [28, 52–56]. Typically, the use of more sensors can provide better estimates of human intent, but there is often a trade-off. For the case of force/torque sensors on the EXO-UL8, that trade-off is wearability, which is crucial for operators with muscular impairments. Chapter 6 explores a method for reducing the number of sensors while retaining comparable transparency.

1.4 Constrained Control

In addition to constraining the admittance control for safety, the allowable motions may be further constrained to enable certain applications, such as virtual object manipulation, which is common in VR-based rehabilitation; and path following, which plays a role in trajectory-based reaching AAN rehabilitation. These seemingly unrelated objectives can be unified by considering them as applications of set stabilization: the allowable motions of the human-robot system are constrained to a subset of the original workspace. For a bimanual exoskeleton, the first interaction constrains the relative pose between the hands to be constant, which is equivalent to respecting the geometry of a rigid object. The second interaction constrains the position of the end effector to be on the one-dimensional path. In each of these scenarios, the desired behavior can be achieved by constraining the configuration space of the human-robot system to a subset of the total space. The system's dynamics are free to move inside the constrained subset, but must be controlled to not leave the subset. The topic of constrained admittance control for rehabilitation applications is explored in Chapter 7.

1.5 EXO-UL8 Exoskeleton

The "robot" aspect of pHRI discussed in this dissertation is primarily implemented on the EXO-UL8 exoskeleton, which serves as the robotic platform for the majority of the experiments. The EXO-UL8 is a bimanual upper limb exoskeleton system developed by the Bionics Lab at the University of California Los Angeles (UCLA) to support research efforts in exoskeleton-based rehabilitation [4–6,57–61]. An exoskeleton robot is a wearable robotic system whose links and joints align with those of the operator, and is able to assist the operator with certain motions. The EXO-UL8 consists of two independently controlled robotic arms with eight independently actuated joints corresponding to those of a human. The exoskeleton is worn by an operator and controlled by the operator's movements through admittance control, which maps the motions of the operator's arms to those of the exoskeleton. Since the EXO-UL8 serves as the underlying hardware that supports many of the aforementioned studies, it is introduced in Chapter 2. The hardware, kinematics, dynamics, control, and implementation are discussed.

1.6 Dissertation Overview and Summary of Contributions

A summary and overview of the chapters in this dissertation are as follows:

- Chapter 1 introduces the main aspects of pHRI explored in this dissertation. An overview and brief background are provided for each section.
- Chapter 2 discusses the EXO-UL8 exoskeleton hardware, kinematics, computed-torque control, and software implementation details. The exoskeleton serves as the platform on which subsequent control methodologies are developed and evaluated.
- Chapter 3 introduces the admittance control methodology and how virtual dynamics are used to generate reference trajectories. A section on sensor fusion discusses how a Kalman filter is used to combine multiple wrenches measured by force/torque sensors on the EXO-UL8 to propagate the virtual dynamics.
- Chapter 4 introduces the primary safety methodology used on both the EXO-UL8 and V-Rex exoskeleton robots. The section discusses how the virtual admittance control dynamics are bounded through the use of soft bounds and hard bounds, which generate restoring forces and emulate infinitely stiff virtual walls, respectively. A multi-arm collision avoidance algorithm is also presented. The overall methodology is then evaluated on both exoskeletons to demonstrate its generalizability.
- Chapter 5 explores the effects of time delay on human-induced instability, which is caused by a combination of neural-muscular and electromechanical delays. The section also introduces how rate-limiting can prevent time delays from destabilizing the system. Experimental results validate the rate-limiter, demonstrating its role as a potential safeguard against time delays.

- Chapter 6 adapts the Kalman filter-based sensor fusion algorithm to provide comparable estimation performance using a strict subset of sensors in an effort to improve wearability of the EXO-UL8. Experimental results demonstrate the approach's feasibility and quantify the impact to transparency using power exchange and operator discomfort as metrics.

- Chapter 7 presents a feedback linearization-inspired method for enabling constrained admittance control, which has direct applications in path-following in AAN robot rehabilitation tasks and VR-based rehabilitation using virtual objects. The constrained admittance control methodology is experimentally verified on both the V-Rex and EXO-UL8 systems in order to demonstrate the methodology and showcase its generalizability.

- Chapter 8 summarizes the main contributions of this work.

CHAPTER 2

EXO-UL8 Exoskeleton System and Control

The EXO-UL8 is a custom, powered, bimanual, redundant, upper-limb, anthropomorphic exoskeleton consisting of two arms, each with seven revolute DoFs and one active gripper DoF [4–6,57,58,62], designed to support research efforts in pHRI and robot-assisted rehabilitation. The EXO-UL8 serves as the robot hardware on which a significant portion of the methods and algorithms within this dissertation are implemented and verified. Thus, this chapter discusses the details of the exoskeleton's mechanics, control, and improvements.

2.1 Kinematics

The arms of the EXO-UL8 are mirrored to correspond to the two arms of a healthy human operator. Most of the joints of exoskeleton correspond to the anatomical joints of a human arm: the first two joints provide the range of motion for should abduction/adduction and shoulder flexion/extension, the third joint corresponds to shoulder interior/exterior rotation, the fourth to elbow flexion/extension, the fifth to forearm pronation/supination, the sixth to wrist extension/flexion, the seventh to wrist radial/ulnar deviation, and the eighth to opening the closing of the hand. All of the joints are revolute to agree with their anatomical counterparts. Figure 2.1 summarizes the kinematic structure of the one of the two arms; the locations of the joints and actuators is independent of the chirality of the arms. Table 2.1 summarizes the anatomical and exoskeleton range of motions for the joints; anatomical values are from [63]. Note that the eighth joint is not considered in this chapter's analysis as it does not affect admittance control of the EXO-UL8's arm.

Joint	DoF Name	Anatomical Limits	EXO-UL8 Limits
1	Shoulder abduction/adduction	$180^\circ/0^\circ$	$90^\circ/0^\circ$
2	Shoulder flexion/extension	$180^\circ/50^\circ$	$90^\circ/10^\circ$
3	Shoulder internal/external rotation	$95^\circ/70^\circ$	$68.75^\circ/15^\circ$
4	Elbow flexion/extension	$140^\circ/0^\circ$	$105^\circ/0^\circ$
5	Forearm pronation/supination	$90^\circ/90^\circ$	$70^\circ/39^\circ$
6	Wrist extension/flexion	$70^\circ/73^\circ$	$45^\circ/29^\circ$
7	Wrist radial/ulnar deviation	$27^\circ/27^\circ$	$29^\circ/30^\circ$

Table 2.1: Joint limits for a typical human arm and the designed joint limits for the EXO-UL8.

Along each of the two arms, three 6-axis force/torque sensors (ATImini 40) are utilized to measure interaction forces with the operator. The first two sensors are located at the upper arm and lower arm, while the third is integrated into the wrist assembly. The locations of these sensors in the kinematic chain is shown in Figure 2.1.

2.1.1 Homogeneous Transformation

Kinematic analysis of the EXO-UL8 is performed using the *product-of-exponentials* (POE) methodology, as described in [40]. POE is advantageous over conventional Denavit-Hartenberg approaches in that the underlying screw theory provides more intuitive geometric interpretation and uses only two reference frames: the body and the spatial frame [40]. For each of the two arms on the EXO-UL8, the origin of the spatial frame is located at the intersection of the three shoulder axes of revolution and oriented such that the x-axis points from the left arm to the right arm, the y-axis points in the forward direction a person faces while wearing the exoskeleton, and the z-axis points upwards, perpendicular from the ground. Figure 2.2 shows the location and orientation of these frames.

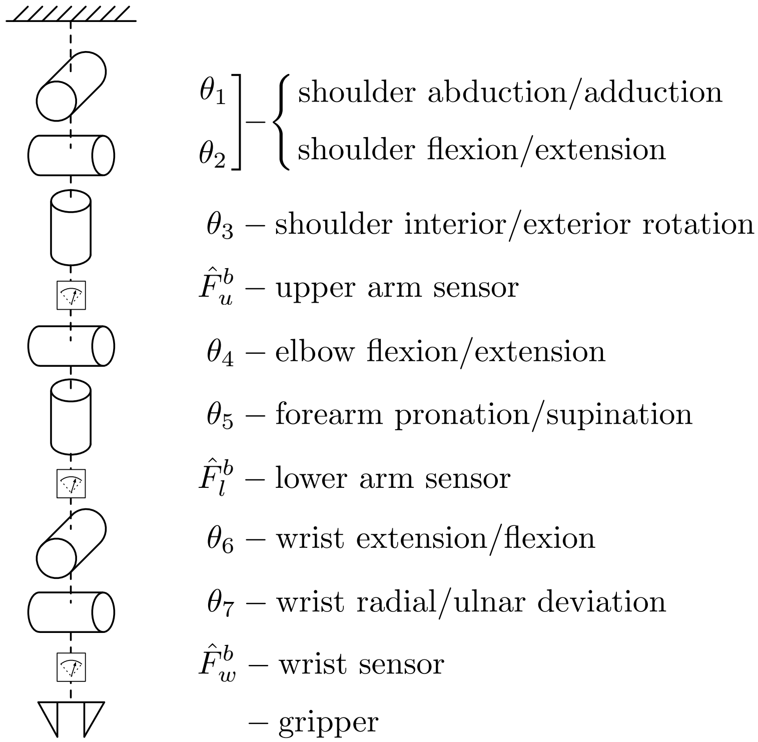


Figure 2.1: Kinematic schematic of one arm on the EXO-UL8 showing the relative locations of all seven revolute joints and three force/torque sensors. The first two joints do not correspond exactly to the anatomical joints. The gripper accounts for the eighth degree-of-freedom.

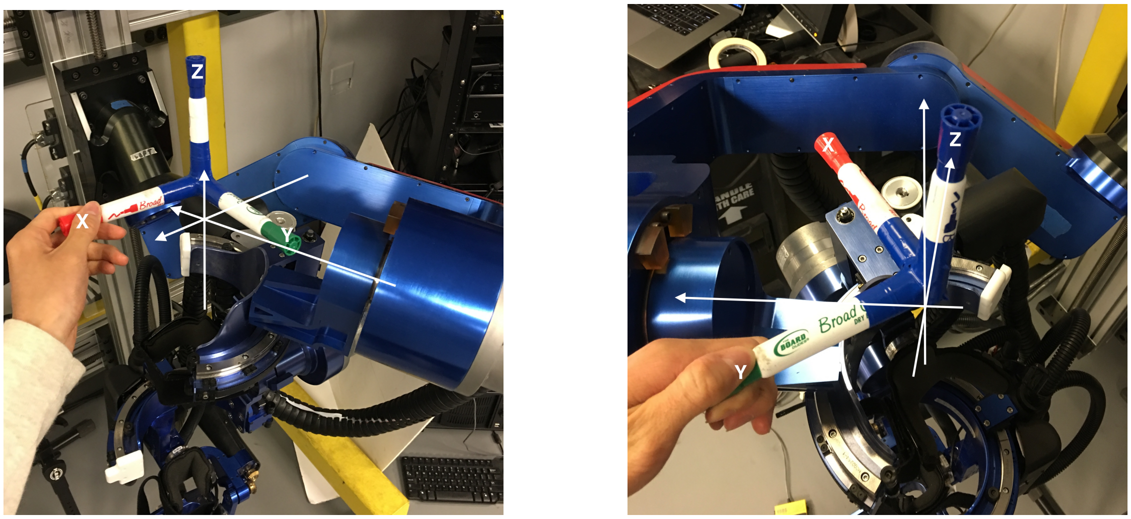
In forward kinematics, the objective is to relate the end-effector's pose to the spatial frame, while being parameterized by joint angles $(\theta_1, \ldots, \theta_7)$. Let $g_{st}(0) \in \mathrm{SE}(3)$, where $\mathrm{SE}(3) \subset \mathbb{R}^{4\times4}$ is the special Euclidean group, represent a transformation to the end-effector frame from the spatial frame at the shoulder with the EXO-UL8 when its joints are in their initial zero position. Then,

$$g_{st}(\theta) = e^{\widehat{\xi_1}\theta_1} \cdots e^{\widehat{\xi_7}\theta_7} g_{st}(0), \tag{2.1}$$

for some joint displacements, $\theta_1, \ldots, \theta_7$. Each ξ_i is a local twist coordinate, and for revolute joints, has the form

$$\xi_i = \begin{bmatrix} -\omega_i \times q_i \\ \omega_i \end{bmatrix}, \tag{2.2}$$

where ω_i is the unit-norm axis of rotation of the joint, and q_i is any point along the axis of



(a) Left arm

(b) Right arm

Figure 2.2: Spatial reference frames and axes of rotation of the first three joints.

rotation. The hat operator maps into a matrix $\widehat{\xi}_i \in \mathfrak{se}(3)$ from the local coordinate $\xi \in \mathbb{R}^6$, where $\mathfrak{se}(3)$ is the Lie algebra of SE(3) at its identity element.

$$\widehat{\xi}_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}^{\wedge} = \begin{bmatrix} \widehat{\omega}_i & v_i \\ 0 & 0 \end{bmatrix}, \tag{2.3}$$

where $\widehat{\omega}_i \in \mathfrak{so}(3)$, the group of 3×3 skew-symmetric matrices and the Lie algebra of SO(3) at its identity element, and

$$\widehat{\omega}_i = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{2.4}$$

where the hat operator here maps local coordinates in $\mathbb{R}^3$ to matrices in $\mathfrak{so}(3)$. Each of the seven transformation matrices in equation (2.1) can be computed using

$$e^{\widehat{\xi}_i \theta_i} = \begin{bmatrix} e^{\widehat{\omega}_i \theta_i} & (I - e^{\widehat{\omega}_i \theta_i})(\omega_i \times v_i) + \omega_i \omega_i^{\top} v_i \theta_i \\ 0 & 1 \end{bmatrix}, \tag{2.5}$$

$$e^{\widehat{\omega}_i \theta_i} = I + \widehat{\omega}_i \sin(\theta_i) + \widehat{\omega}_i^2 (1 - \cos(\theta_i)). \tag{2.6}$$

The axes of rotation of the joints and points along the axes of rotation in the spatial frame are given in Tables 2.2 and 2.3, respectively. Note that these vectors and points are when the exoskeleton is in its initial zero-position configuration.

Joint	Left Arm Rotation Axes (m)	Right Arm Rotation Axes (m)
1	$[0.707107, 0.521334, 0.477714]$	$[0.707107, -0.521334, -0.477714]$
2	$[0.593426, -0.804889, 0]$	$[0.593426, 0.804889, 0]$
3	$[0, 0, -1]$	$[0, 0, 1]$
4	$[1, 0, 0]$	$[1, 0, 0]$
5	$[0, 0, -1]$	$[0, 0, 1]$
6	$[0, 1, 0]$	$[0, -1, 0]$
7	$[1, 0, 0]$	$[1, 0, 0]$

Table 2.2: Axes of rotation for each joint.

Joint	Point along Axes (m)
1	$[0, 0, 0]$
2	$[0, 0, 0]$
3	$[0, 0, 0]$
4	$[0, 0, -0.3036]$
5	$[0, 0, -0.3036]$
6	$[0, 0, -0.5803]$
7	$[0, 0, -0.5803]$

Table 2.3: Point along each axes of rotation for each joint.

2.1.2 Spatial Manipulator Jacobian

The velocity of the end-effector (expressed in the spatial coordinate frame) can be written as a function of the joint positions and velocities:

$$\hat{v}_s^{sp} = \dot{g}_{st}(\theta) g_{st}^{-1}(\theta). \tag{2.7}$$

The time derivative can be carried out, and equation (2.7) can be rewritten as

$$v_s^{sp} = J_s^{sp}(\theta)\dot{\theta}, \tag{2.8}$$

where $J_s^{sp}(\theta)$ is defined as the spatial manipulator Jacobian. The Jacobian is computed as

$$\begin{aligned} J_s^{sp}(\theta) &= \left[\left(\frac{\partial g_{st}}{\partial \theta_1} g_{st}^{-1} \right)^{\vee} \cdots \left(\frac{\partial g_{st}}{\partial \theta_n} g_{st}^{-1} \right)^{\vee} \right], &\quad (2.9)\\ &= [\xi_1 \; \xi_2^{'} \cdots \xi_n^{'}], &\quad (2.10) \end{aligned}$$

where

$$\xi_i^{'} = \mathrm{Ad}_{\left(e^{\widehat{\xi}_1 \theta_1} \cdots e^{\widehat{\xi}_{i-1}\theta_{i-1}}\right)} \xi_i. \tag{2.11}$$

Note that the adjoint for a homogeneous transformation $g = \left[\begin{smallmatrix} R & p \\ 0 & 1 \end{smallmatrix}\right] \in \mathrm{SE}(3)$ is given by

$$\mathrm{Ad}_g = \begin{bmatrix} R & \widehat{p}R \\ 0 & R \end{bmatrix}. \tag{2.12}$$

2.1.3 Force Transformation

Contact force between the human and the exoskeleton is measured by the ATImini 40 force sensor modules located at the upper arm, the lower arm, and the wrist, as shown in Figure 2.3. Each of these sensors provides a wrench measurement (3 DoF for force, 3 DoF for torque). These measurements can be transformed into torques applied to the exoskeleton's joints by using the spatial manipulator Jacobian. First, the measured wrenches are transformed from the sensor's frame to the spatial frame through

$$F_s^{sp} = \mathrm{Ad}_{g_s^{-1}(\theta)}^{\top} F_s^b, \tag{2.13}$$

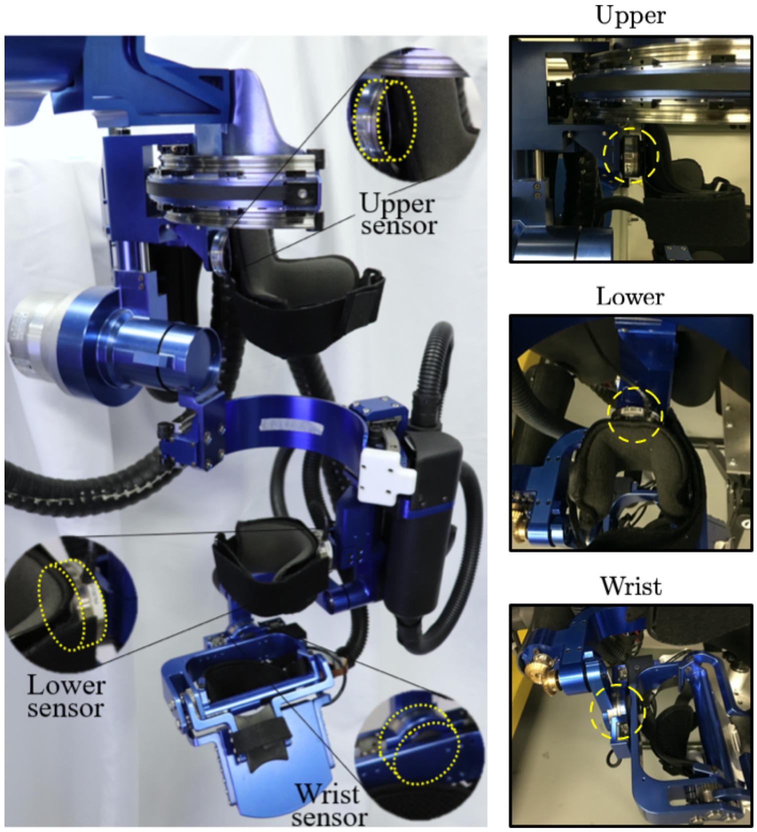


Figure 2.3: Each arm of the EXO-UL8 incorporates three 6-axis ATImini40 force/torque sensors located at the upper arm, the lower arm, and the wrist assembly.

where $F_s^{sp} \in \mathbb{R}^6$ expresses the equivalent wrench in the spatial frame. The transformed wrenches, F_s^{sp}, are then mapped to joint torques $\Gamma_s \in \mathbb{R}^7$ with the spatial manipulator Jacobian:

$$\Gamma_s = J_s^{sp}(\theta)^\top F_s^{sp}. \tag{2.14}$$

Each of the sensors contributes a torque vector ($\Gamma_u \in \mathbb{R}^3, \Gamma_l \in \mathbb{R}^5, \Gamma_w \in \mathbb{R}^7$) to the admittance controller. Note that the dimensions of the spatial manipulator Jacobian are different for each sensor due to each sensor being located at a different position along the kinematic chain, as shown in Figure 2.1. Table 2.4 gives the physical locations of the sensors.

Sensor	Right Side Relative Location (m)	Left Side Relative Location (m)
Upper	$[0.0467, 0, -0.2069]$	$[-0.0467, 0, -0.2069]$
Lower	$[0, 0, -0.5134]$	$[0, 0, -0.5134]$
Wrist	$[0.058, 0, -0.6528]$	$[-0.058, 0, -0.6528]$

Table 2.4: Location of force sensors on the right and left sides.

2.2 Dynamics

The dynamical equations of motion can be formulated using the Lagrangian, as detailed in [40]. Similar to the previous section, it suffices to consider only one of the arms. In order to calculate the total kinetic energy of the arm, let a coordinate frame, L_i, be fixed to the center of mass of the i^{th} link and oriented with the principle axes of inertia. Then the homogeneous transformation from the base frame to the link frame is given as:

$$g_{sl_i}(\theta) = e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_i\theta_i} g_{sl_i}(0), \tag{2.15}$$

where $i \in \{1, \ldots, 7\}$. The corresponding body Jacobian is given as:

$$J^b_{sl_i}(\theta) = \begin{bmatrix} \xi_1^\dagger & \cdots & \xi_i^\dagger & 0 & \cdots & 0 \end{bmatrix}, \tag{2.16}$$

where $\xi_j^\dagger$ is the j^{th} instantaneous joint twist relative to the i^{th} frame, and is given as:

$$\xi_j^\dagger = \mathrm{Ad}^{-1}_{\left(e^{\widehat{\xi}_j\theta_j} \cdots e^{\widehat{\xi}_i\theta_i} g_{sl_i}(0)\right)} \xi_j, \quad j \leq i. \tag{2.17}$$

The body Jacobian allows the body velocity of the i^{th} link's center of mass to be written as:

$$V^b_{sl_i} = J^b_{sl_i}(\theta)\dot{\theta}. \tag{2.18}$$

Next, let the generalized inertia for the link be:

$$\mathcal{M}_i = \begin{bmatrix} m_i I & 0 \\ 0 & \mathcal{I}_i \end{bmatrix}, \tag{2.19}$$

where m_i is the link's mass, I is the 3×3 identity matrix, and $\mathcal{I}_i$ is the link's diagonal inertia tensor. Note that $\mathcal{M}_i$ is diagonal only because L_i is attached to the center of the mass of the link and aligned with the principle axes of inertia. Finally, the kinetic energy of the link is:

$$T_i(\theta, \dot{\theta}) = \frac{1}{2}(V^b_{sl_i})^\top \mathcal{M}_i V^b_{sl_i}, \tag{2.20}$$

$$= \frac{1}{2}\dot{\theta}^\top J^b_{sl_i}(\theta)^\top \mathcal{M}_i J^b_{sl_i}(\theta)\dot{\theta}. \tag{2.21}$$

By defining $M(\theta) := \sum_i J^b_{sl_i}(\theta)^\top \mathcal{M}_i J^b_{sl_i}(\theta)$, the total kinetic energy of the manipulator can be written as:

$$T(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^\top M(\theta)\dot{\theta}, \tag{2.22}$$

where $M(\theta)$ is known as the manipulator inertia matrix. The potential energy for the manipulator is solely due to the effects of gravity. In this case, the potential energy for the i^{th} link is:

$$V_i(\theta) = m_i g h_i(\theta), \tag{2.23}$$

where $h_i(\theta)$ is the height of the link's center of mass, and g is the gravitational constant. By defining $V(\theta) = \sum_i V_i(\theta)$, the Lagrangian of the manipulator can be written as:

$$L(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^\top M(\theta)\dot{\theta} - V(\theta). \tag{2.24}$$

2.2.1 Equations of Motion

Let u_i be the total torque acting on the i^{th} joint. Then, the equations of motion are:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L(\theta, \dot{\theta})}{\partial \dot{\theta}_i} - \frac{\partial L(\theta, \dot{\theta})}{\partial \theta_i} = u_i, \tag{2.25}$$

where $L(\theta, \dot{\theta})$ is the Lagrangian from equation (2.24), and $i \in \{1, \ldots, 7\}$. Carrying out the differentiation, the equations of motion become:

$$\sum_j M_{ij}(\theta)\ddot{\theta}_j + \sum_{j,k}\left(\frac{\partial M_{ij}(\theta)}{\partial \theta_k}\dot{\theta}_j\dot{\theta}_k - \frac{1}{2}\frac{\partial M_{kj}(\theta)}{\partial \theta_i}\dot{\theta}_k\dot{\theta}_j\right) + \frac{\partial V(\theta)}{\partial \theta_i} = u_i. \tag{2.26}$$

The Christoffel symbols of the first kind, Γ_{ijk}, for the manipulator inertia matrix are commonly used to simplify the equations of motion. They are given as:

$$\Gamma_{ijk} = \frac{1}{2}\left(\frac{\partial M_{ij}(\theta)}{\partial \theta_k} + \frac{\partial M_{ik}(\theta)}{\partial \theta_j} - \frac{\partial M_{kj}(\theta)}{\partial \theta_i}\right). \tag{2.27}$$

Applying the substitution, equation (2.26) then becomes:

$$\sum_j M_{ij}(\theta)\ddot{\theta}_j + \sum_{j,k} \Gamma_{ijk}\dot{\theta}_j\dot{\theta}_k + \frac{\partial V(\theta)}{\partial \theta_i} = u_i. \tag{2.28}$$

To write the equations of motion in vector form, a couple more definitions are made. Let $N \in \mathbb{R}^7$ represent the conservative forces and be defined as:

$$N_i(\theta) = \frac{\partial V(\theta)}{\partial \theta_i}. \tag{2.29}$$

Furthermore, let $C(\theta, \dot{\theta}) \in \mathbb{R}^{7\times7}$ be a square matrix with elements:

$$C_{ij}(\theta, \dot{\theta}) = \sum_k \Gamma_{ijk}\dot{\theta}_k. \tag{2.30}$$

Then, the equations of motion can be written concisely as:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta) = u, \tag{2.31}$$

where $M(\theta) \in \mathbb{R}^{7\times7}$ is the manipulator inertia matrix, $C(\theta, \dot{\theta}) \in \mathbb{R}^{7\times7}$ is the manipulator Coriolis matrix, $N(\theta) \in \mathbb{R}^7$ is the conservative force vector, and $u \in \mathbb{R}^7$ is the vector of joint torques. Equation (2.31) conveniently represents the dynamics of the manipulator in vector form, and will be used for subsequent analysis.

2.2.2 EXO-UL8 Parameters

The parameters for the EXO-UL8 are calculated using the Mass Properties tool from SOLIDWORKS. Material densities and component masses are filled into the SOLIDWORKS model according to their datasheet values. The parameters for the right arm of the EXO-UL8 are given in Tables 2.5 and 2.6. All coordinate-dependant quantities are expressed in the base frame.

Since the left arm is a mirror copy of the right arm, its dynamical parameters can be computed from those of the right arm. However, since the left arm is mirrored in the plane normal to the x-axis, care must be taken to ensure that only certain values have their sign flipped. The center of mass of each link i on the left arm can be computed by simply negating the x-component for the corresponding right arm link. Intrinsic properties, such as mass or principle inertias, do not change between the arms. For the principle axes of inertia, flip the direction of the first principle axis. Then, negate only the x-coordinates of the three axes. In order words:

$$\left\{ \begin{bmatrix} a_x^{(1)} \\ a_y^{(1)} \\ a_z^{(1)} \end{bmatrix}, \begin{bmatrix} a_x^{(2)} \\ a_y^{(2)} \\ a_z^{(2)} \end{bmatrix}, \begin{bmatrix} a_x^{(3)} \\ a_y^{(3)} \\ a_z^{(3)} \end{bmatrix} \right\} \longrightarrow \left\{ \begin{bmatrix} a_x^{(1)} \\ -a_y^{(1)} \\ -a_z^{(1)} \end{bmatrix}, \begin{bmatrix} -a_x^{(2)} \\ a_y^{(2)} \\ a_z^{(2)} \end{bmatrix}, \begin{bmatrix} -a_x^{(3)} \\ a_y^{(3)} \\ a_z^{(3)} \end{bmatrix} \right\}. \tag{2.32}$$

Since the principle axes must form a right-handed coordinate frame located at the center of mass, any two of the axes can flip and still yield the same inertia properties.

Joint	Link Masses (kg)	Center of Mass (m)	Principle Inertias ($\mathrm{kg} \cdot \mathrm{m}^2$)
1	14.16052	$[0.23727, 0.13825, -0.06131]$	$[0.09249, 0.49091, 0.54456]$
2	1.90092	$[0.05823, 0.05051, -0.09990]$	$[0.00517, 0.01295, 0.01560]$
3	7.70825	$[0.18034, -0.02335, -0.26417]$	$[0.02499, 0.05136, 0.05415]$
4	1.30118	$[0.01520, -0.03836, -0.37523]$	$[0.00235, 0.00759, 0.00841]$
5	2.29959	$[-0.02194, -0.10862, -0.46951]$	$[0.00360, 0.00925, 0.01084]$
6	0.74245	$[0.09193, -0.05096, -0.58021]$	$[0.00051, 0.00163, 0.00205]$
7	0.85143	$[0.01190, 0.04524, -0.65505]$	$[0.00120, 0.00019, 0.00180]$

Table 2.5: Link masses, centers of mass, and principle inertias for the right arm of the EXO-UL8.

2.3 Computed Torque Control

2.3.1 Control Output

Given the equations of motion from equation (2.31), the computed torque method of [40] is implemented for the EXO-UL8. Given a desired joint trajectory, θ^{ref}, along with its time-derivatives, $\dot{\theta}^{\text{ref}}, \ddot{\theta}^{\text{ref}}$, the joint torques are set as:

$$u = M(\theta)(\ddot{\theta}^{\text{ref}} + K_d \dot{e} + K_p e) + C(\theta, \dot{\theta})\dot{\theta} + N(\theta), \tag{2.33}$$

where $e = \theta^{\text{ref}} - \theta$ and $\dot{e} = \dot{\theta}^{\text{ref}} - \dot{\theta}$, and K_p, K_d are constant gain matrices. The computed torque controller essentially tries to cancel out the nonlinearities associated with the Coriolis matrix and the conservative force vector. Substituting equation (2.33) into equation (2.31), the closed-loop dynamics become:

$$M(\theta)(\ddot{e} + K_d \dot{e} + K_p e) = 0, \tag{2.34}$$

where $\ddot{e} = \ddot{\theta}^{\text{ref}} - \ddot{\theta}$. Since $M(\theta)$ is positive definite by construction, the error dynamics simplify to:

$$\ddot{e} + K_d \dot{e} + K_p e = 0, \tag{2.35}$$

Joint	First Principle Axes of Inertia	Second Principle Axes of Inertia
1	$[-0.61357, 0.62841, 0.47816]$	$[-0.65679, -0.74229, 0.13275]$
2	$[0.08146, 0.83151, 0.54951]$	$[-0.72090, 0.42988, -0.54361]$
3	$[-0.40235, -0.31316, 0.86026]$	$[0.05348, 0.93003, 0.36357]$
4	$[-0.94455, -0.29870, -0.13638]$	$[0.32750, -0.88704, -0.32543]$
5	$[-0.09139, -0.03444, 0.99522]$	$[0.41615, 0.90663, 0.06959]$
6	$[-0.18183, 0.98333, -0.00219]$	$[-0.98326, -0.18185, -0.01116]$
7	$[0.19916, 0.89843, 0.39135]$	$[-0.51325, 0.43582, -0.73935]$

Table 2.6: The first two principle axes of inertia about the centers of mass for each link of the right arm of the EXO-UL8 is given. The third axis can be directly computed by taking the cross-product.

which is a stable linear system as long as $K_d > 0$ and $K_p > 0$ (positive definite). The computed torque input can be broken down into the sum of two components:

$$u = \underbrace{M(\theta)\ddot{\theta}^{\text{ref}} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta)}_{\text{feedforward}} + \underbrace{M(\theta)(K_d\dot{e} + K_p e)}_{\text{feedback}}. \tag{2.36}$$

The feedback gains on the error terms are for stability of the error dynamics, while the feedforward component is necessary for cancelling the nonlinearity of the system dynamics and decoupling the error dynamics from the state. Since the nonlinearities cannot be perfectly compensated in practice, the feedback portion is necessary.

Note that the feedforward component $\ddot{\theta}^{\text{ref}}$ is also necessary in order to produce the closed-loop error dynamics of equation (2.35). If a reference acceleration is not generated, i.e. $\ddot{\theta}^{\text{ref}} = 0$, then the closed-loop error dynamics become:

$$\ddot{\theta} + K_d\dot{e} + K_p\ddot{e} = 0. \tag{2.37}$$

In the Laplace domain, the transfer function matrix for these dynamics is of the form:

$$(e/\theta)(s) = -s^2(sK_d + K_p)^{-1}, \tag{2.38}$$

which resembles a high-pass filter. Hence, the error becomes coupled with the state in the absence of feedforward reference acceleration.

2.3.2 State Measurement

Joint positions, θ, can be directly measured at a frequency of 1kHz by the optical encoders at each joint. On the other hand, the joint velocities, $\dot{\theta}$, must be estimated. Traditional linear Kalman filter-based estimation strategies are inadequate due to the nonlinear equations of motion of equation (2.31).

For practical implementation feasibility, the joint velocities are numerically differentiated and then passed through a 4^{th} order Butterworth low-pass filter with corner frequency of 30Hz. The transfer function for each channel of the filter was designed as:

$$H_i(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}, \tag{2.39}$$

where the coefficients given in Table 2.7 are calculated with MATLAB. The discrete time filter was realized with the Direct form II implementation in software.

Index	Numerator (b)	Denominator (a)
0	$6.238\,698\,354\,854\,577\,1 \times 10^{-5}$	1.0
1	$2.495\,479\,341\,941\,830\,9 \times 10^{-4}$	$-3.507\,786\,207\,390\,781\,3$
2	$3.743\,219\,012\,912\,746\,3 \times 10^{-4}$	$4.640\,902\,412\,686\,704\,9$
3	$2.495\,479\,341\,941\,830\,9 \times 10^{-4}$	$-2.742\,652\,821\,120\,370\,5$
4	$6.238\,698\,354\,854\,577\,1 \times 10^{-5}$	$6.105\,348\,075\,612\,233\,6 \times 10^{-1}$

Table 2.7: Discrete time filter coefficients for a 4^{th} order Butterworth low-pass filter.

2.3.3 Friction Compensation

Friction is compensated for the Maxon motor actuators used in joints 3, 5, 6, and 7. Although the feedback controller is able to overcome the effects of friction, explicitly compensating for the friction can improve tracking performance. Since modeling friction is a complex problem, only the most significant component, static friction, is modeled and compensated. Static friction is modeled as an additional torque that acts in the opposite direction as the joint's angular velocity. Thus, the friction torque for the i^{th} joint is modeled as:

$$\tau_{f,i}(\dot{\theta}_i) = \tau_{f,i}^{\text{static}} \operatorname{sgn}(\dot{\theta}_i), \tag{2.40}$$

where sgn is the signum function and $\tau_{f,i}^{\text{static}}$ is an empirically determined constant. Figure 2.4 shows the friction model, and Table 2.8 lists the values of $\tau_{f,i}^{\text{static}}$.

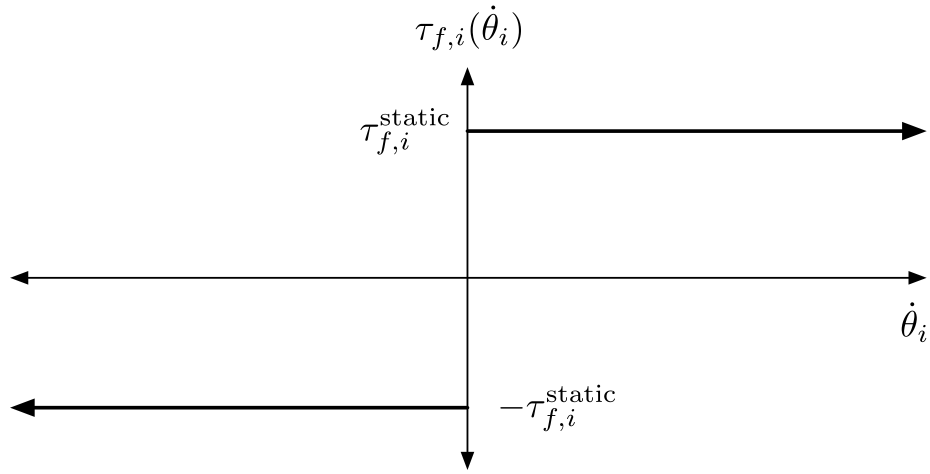


Figure 2.4: Static friction model used for friction compensation on the Maxon motor joints.

For the Harmonic drives of joints 1, 2, and 4, a similar friction model consisting of both static and viscous friction forces is used. To identify the friction torque and velocity relationship, the motor is driven at constant velocities while the current is measured using the vendor-provided Harmonic drive controller. The current is then converted to an estimated torque using the motor's torque constant. Results are plotted for the SHA25 motor (joint 4) in Fig. 2.5.

Joint	$\tau_{f,i}^{\text{static}}$ (Nm)
3	12.0
5	15.0
6	25.0
7	10.0

Table 2.8: Empirically determined constants for modeling static friction on the Maxon motor joints on the EXO-UL8.

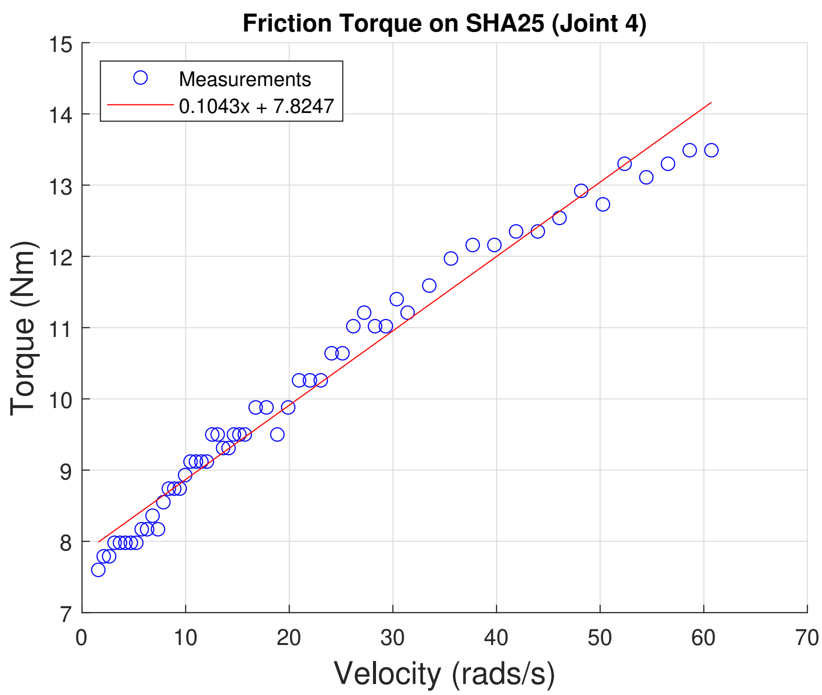


Figure 2.5: Measured relationship between motor torque and velocity, and a least-squares curve-fit to an affine model.

In general, the friction torque to velocity relationship for Harmonic drives is complex and can even contain hysteresis [64]. To simplify the control, an affine model is fit to the data, in the sense of least-squares. The Stribeck effect is not included due to the limited resolution of the current sensing hardware. The resulting model is shown in Fig. 2.6. Note that the static friction component of Fig. 2.4 is replaced by a sloped line so that the torque relationship remains continuous. The model is subsequently used in the computed torque control of the exoskeleton.

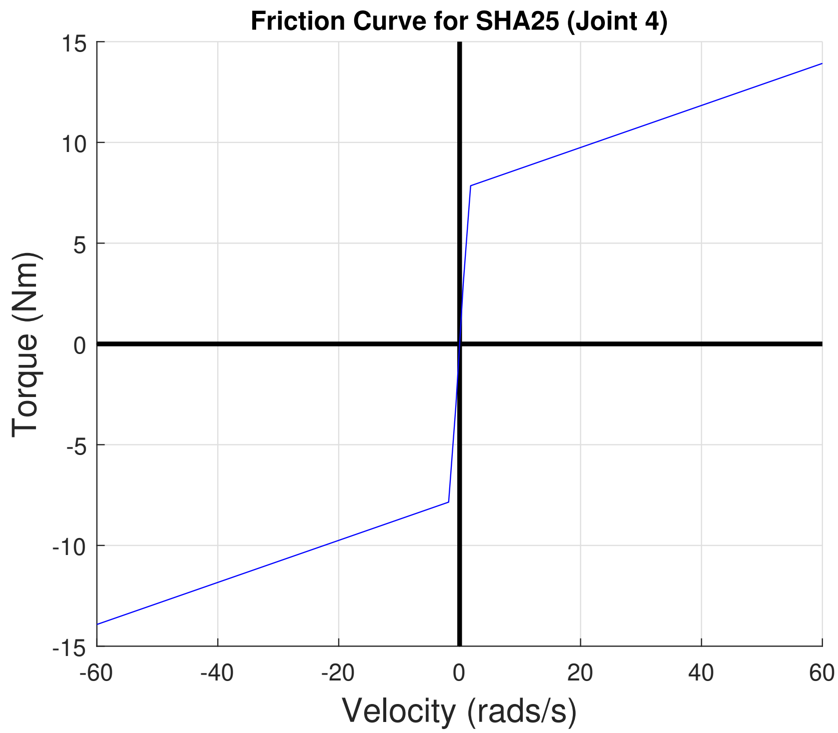


Figure 2.6: Friction model for the SHA25 Harmonic drive used on joint 4 (elbow flexion/extension).

The computed torque control input of equation (2.33) is then augmented with the velocity-dependent static friction:

$$u = M(\theta)(\ddot{\theta}^{\text{ref}} + K_d\dot{e} + K_p e) + C(\theta, \dot{\theta})\dot{\theta} + N(\theta) + \tau_f(\dot{\theta}). \tag{2.41}$$

2.4 Symbolic Dynamics

Implementing the controller of equation (2.41) requires computing the matrices $M(\theta)$, $C(\theta, \dot{\theta})$, and $N(\theta)$ in real-time since they are dependent on the joint position, and also angular velocity in the case of $C(\theta, \dot{\theta})$. Furthermore, the state space is too large to pre-compute these matrices to store in a look-up table. Thus, they must be computed in real-time, and efficiently enough to meet the 1kHz timing requirement of the control loop.

While these matrices can be computed numerically following the steps in subsection 2.1.2 and section 2.2, such a procedure can be inefficient and is not immune to potential issues from floating point numbers. As a result, closed-form expressions parameterize by the state $(\theta, \dot{\theta})$ are symbolically generated using Sympy (version 1.6.1) [65] for Python (version 3.7.6). The package was selected for its support of common subexpression elimination, which allows for complicated symbolic expressions to be broken down into smaller subexpressions, assigned to intermediate variables, and then used in subsequent expressions. The included `C11CodePrinter` class was extended to generate `C` code from the symbolic expressions. The code snippet below shows a sample of the generated code.

```
void manipulatorInertiaMatrix(const double* M, const double* position){
    const double x0 = M_SQRT2;
    const double x1 = cos(position[0]);
    const double x2 = 0.73611111111111116*M_PI;
    const double x3 = sin(x2);
    const double x4 = x1 - 1;
    const double x5 = x3*x4;
    const double x6 = 0.11630055367992947*x5;
    const double x7 = cos(x2);
    const double x8 = x4*x7;
    const double x9 = sin(position[0]);
```

Although several thousand intermediate variables are generated for complex expressions, such as the Coriolis matrix, modern C/C++ optimizing compilers will still produce efficient code. The runtime of using symbolic expressions is often significantly faster than numerically computing all the matrices. Generating all of symbolic expressions for all the matrices and then outputting `C`-style code takes approximately 6 minutes on a 3.1GHz Quad-Core Intel i7 with 16GB of RAM. Note that the code only needs to be generated whenever the mass properties of the EXO-UL8 change.

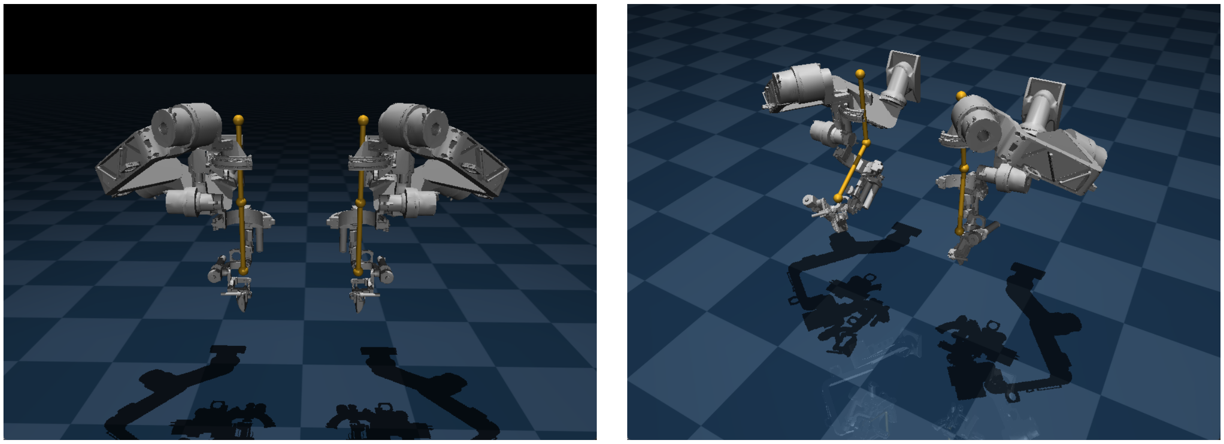


Figure 2.7: Dynamical simulation of the EXO-UL8 is implemented using MuJoCo. The simulator is used throughout during development and verification.

2.5 Verification

2.5.1 Simulation

Dynamical simulation of the EXO-UL8 is performed using MuJoCo [66], as shown in Figure 2.7. A hardware model of the EXO-UL8 is created from the inertia properties of 2.2.2 and visualized with STL outputs from the SOLIDWORKS model. The equations of motion are implemented internally in MuJoCo, and are used to corroborate both the equations of motion derived from the Lagrangian formulation and the symbolic implementation. The MuJoCo-based simulator is also used to verify functionality logic, such as switching between different controllers and references, data logging, visualization, and so on. It has proved to be an invaluable tool throughout development and verification of all subsequent methods.

2.5.2 Hardware Testing

The computed torque controller was applied to the Maxon motor-actuated joints and compared to the former PD controllers. The reference and measured trajectories for the joints are shown in Figures 2.8, 2.9, 2.10, and 2.11. Results indicate that tracking performance has improved.

To quantitatively assess the performance, the RMS of the error signal is calculated. For an error signal, $e[k]$, of length N, the RMS is defined as:

$$\text{RMS} = \sqrt{\frac{1}{n}\sum_{k=1}^{N} e[k]^2}. \tag{2.42}$$

The RMS quantifies the magnitude of the error signal and provides a comparable metric independent of the signal duration. The RMS values for the error signals are reported in Table 2.9.

Joint	PD Controller	Model-based Controller
3	2.297°	0.302°
5	0.788°	0.417°
6	0.964°	0.208°
7	1.126°	0.237°

Table 2.9: Error RMS values comparing the tracking performance between the original PD controller and the computed torque controller. Error values are reported in degrees.

The largest improvements can be seen in joint 3. Since this joint moves the largest mass due to its higher position in the kinematic chain compared to the other joints, gravity compensation benefits this joint the most. Furthermore, the tracking errors in the former PD controller for joint 3 are skewed towards positive errors, which happens during shoulder interior rotation with the arm extended in front. In this scenario, gravity acts against tracking the reference, so the skew in errors are expected. This is not the case in the model-based controller due to its gravity compensation. The other joints are loaded with less mass, so the performance improvements are relatively smaller.

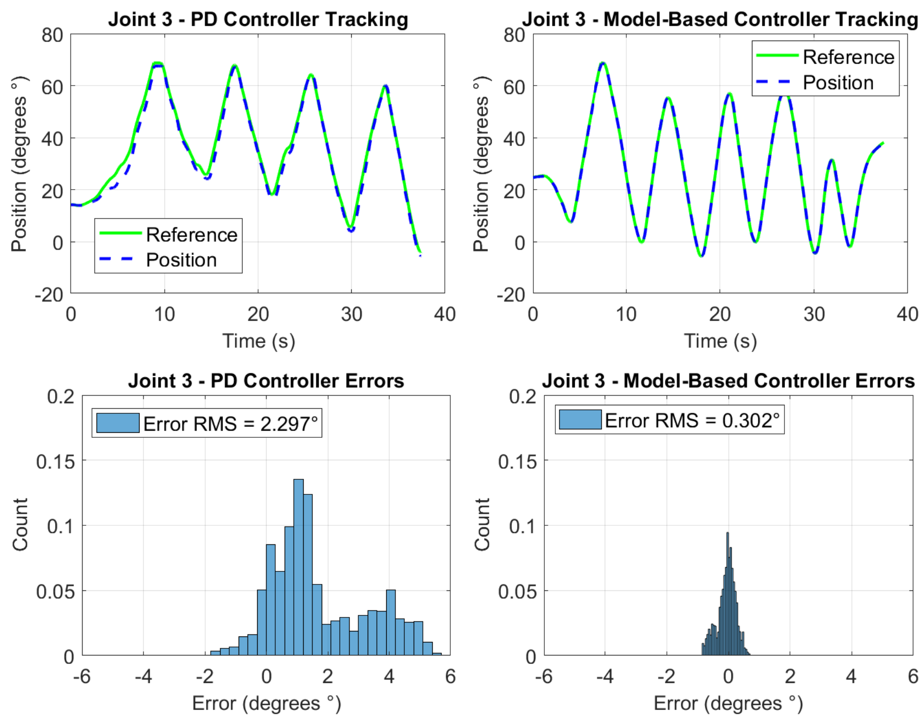


Figure 2.8: Tracking performance for joint 3: shoulder interior/exterior rotation.

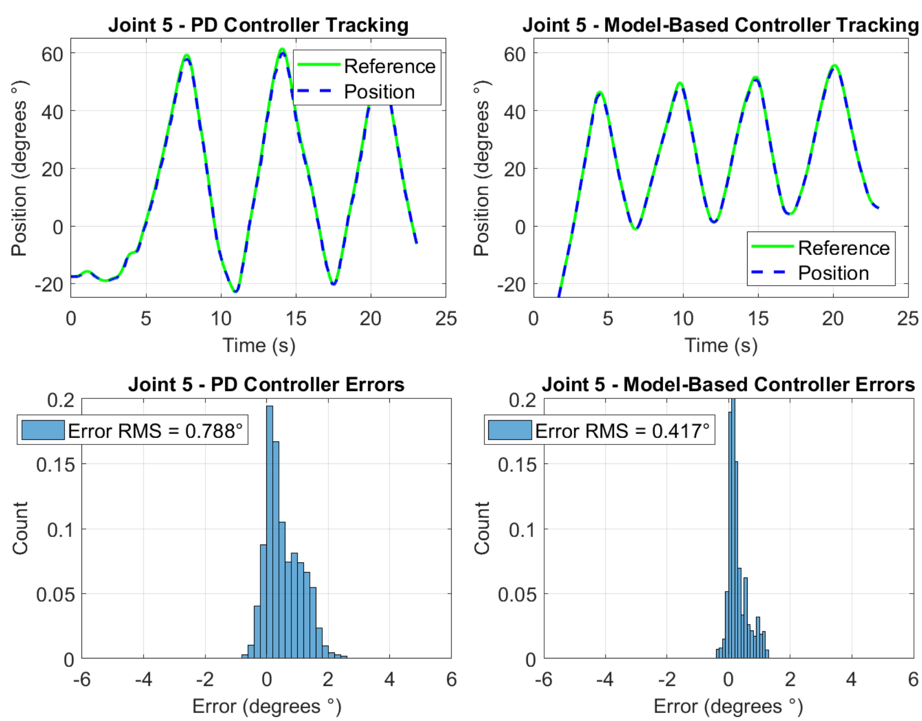


Figure 2.9: Tracking performance for joint 5: forearm supination/pronation.

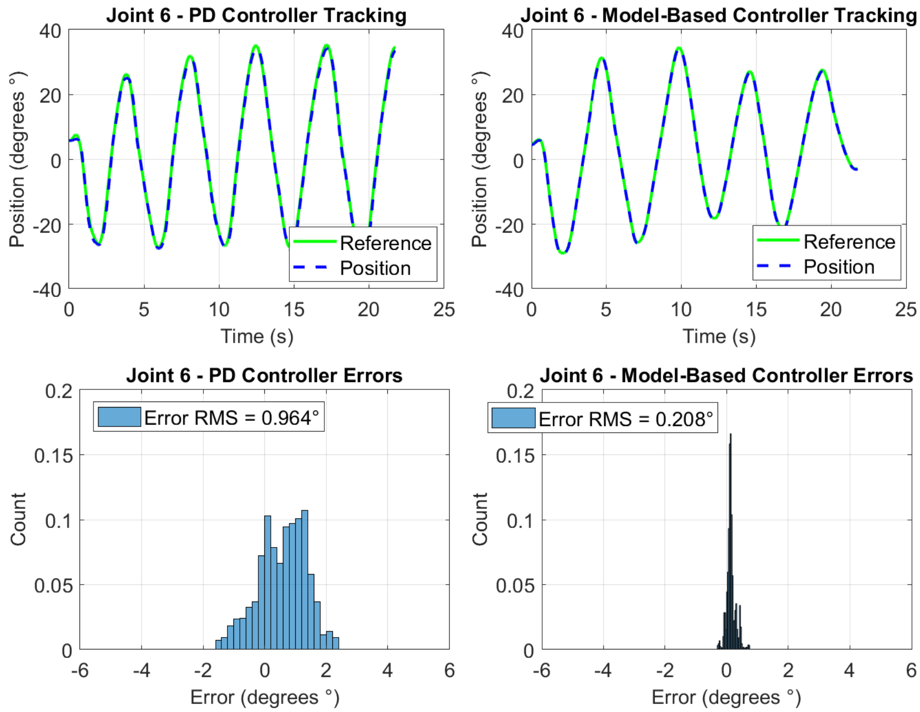


Figure 2.10: Tracking performance for joint 6: wrist flexion/extension.

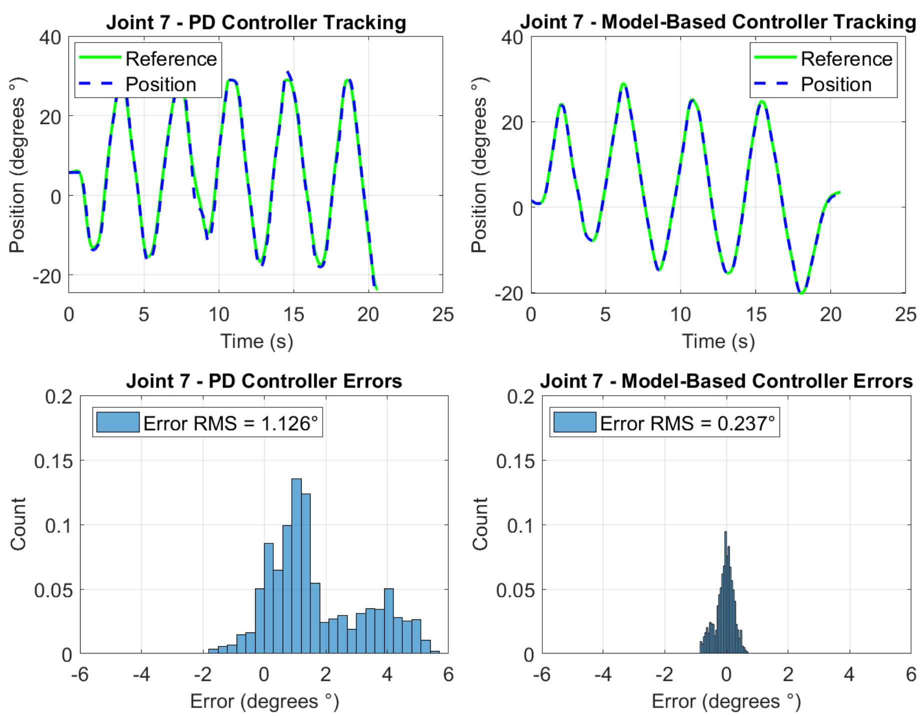


Figure 2.11: Tracking performance for joint 7: wrist ulnar/radial deviation.

2.6 Hardware Updates

Actuators on the EXO-UL8 consist of brushless DC motors from Maxon Motors (joints 3, 5, 6, 7) in current-control mode enabled by Escon 50-5 motor controllers, and harmonic drives from the company Harmonic Drive (joints 1, 2, 4) operated in either current- or velocity-control mode. For either actuator type, the input signal is passed as an analog voltage (-10V to 10V), generated from a DAC on the industrial computer running the control loop. The existing DAC is the PCI-DAC6702 from Measurement Computing, capable of generating 16-bit analog output at an aggregate frequency of 1KHz for all the channels. Since the data output is bottle-necked by the DAC, the I/O operations were initially asynchronous to the main control loop. While this was fine for operation in the past, a hardware update is necessary in order to improve joint space tracking performance, which is necessary for high transparency.

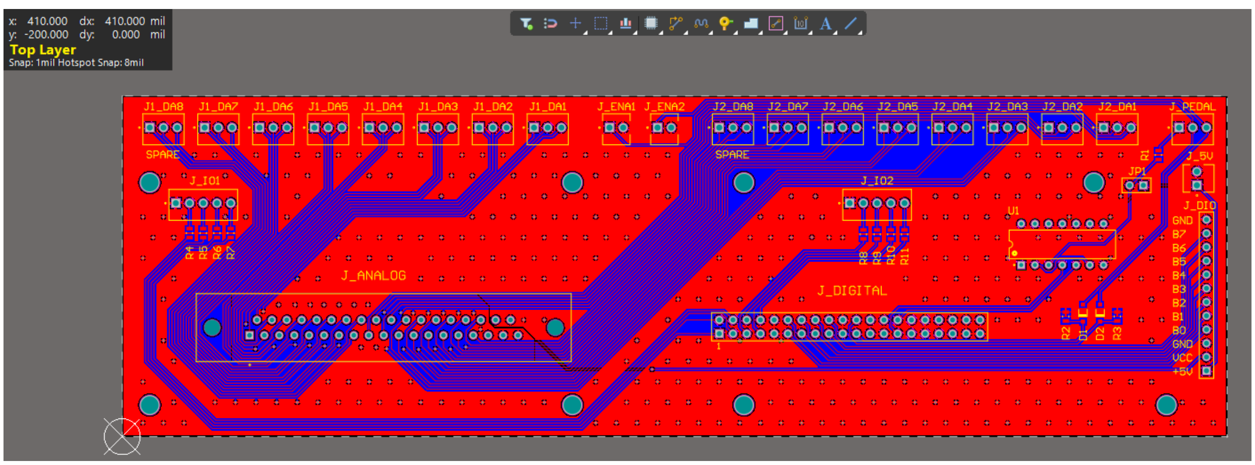


Figure 2.12: Design of the updated analog signal distribution circuit in order to accommodate the new PCIe-DA16-16 DAC

.

To this end, the card is replaced by the PCIe-DA16-16 DAC from Acces I/O Products, which is capable of updating outputs at a frequency of at least 1KHz for each channel independently. Upgrading the DAC card also required a redesign of the analog signal distribution circuit board, which was completed using Altium Designer, as shown in Figure 2.12. The

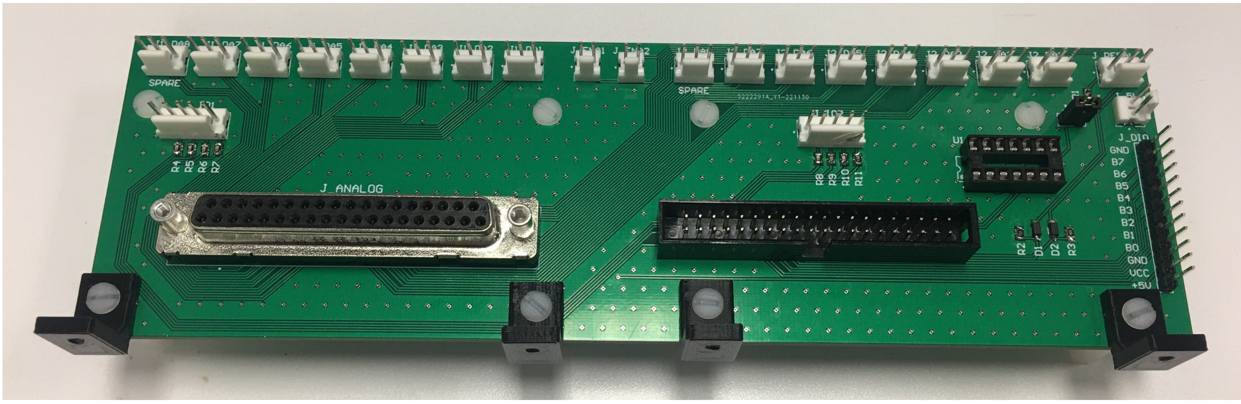


Figure 2.13: Built and assembled analog signal distribution circuit board.

.

circuit board is subsequently built with JLC-PCB, manually assembled as shown in Figure 2.13, and installed into the EXO-UL8 electronics box. The upgrade enabled the controller software and voltage output to run synchronously at 1KHz, which improved tracking performance and facilitated subsequent software development on the EXO-UL8.

2.7 Software

To ensure consistent performance while being compatible with device drivers, the core control loop utilizes the Multimedia Timer running on Windows 10. Although the operating system is not real-time, the use of the Multimedia Timer allowed for a consistent 1KHz loop in which the admittance controller and computed-torque controllers are implemented. All aspects of the software are written in C++17 and compiled with MSVC provided through Visual Studio 2019. Third party tools and libraries used include:

- CMake - build system,
- Eigen - template library for matrices, vectors, and linear algebra,
- QT6 - graphical user interface,
- Boost - Network UDP operations,

- MuJoCo - Dynamical simulator and visualizer,
- OSQP - quadratic program solver,
- {fmt} - printing and formatting,
- nlohmann/JSON - JSON parsing.

Version control was handled by `git` and hosted on GitHub. A screenshot of the graphical user interface is shown in Figure 2.14.

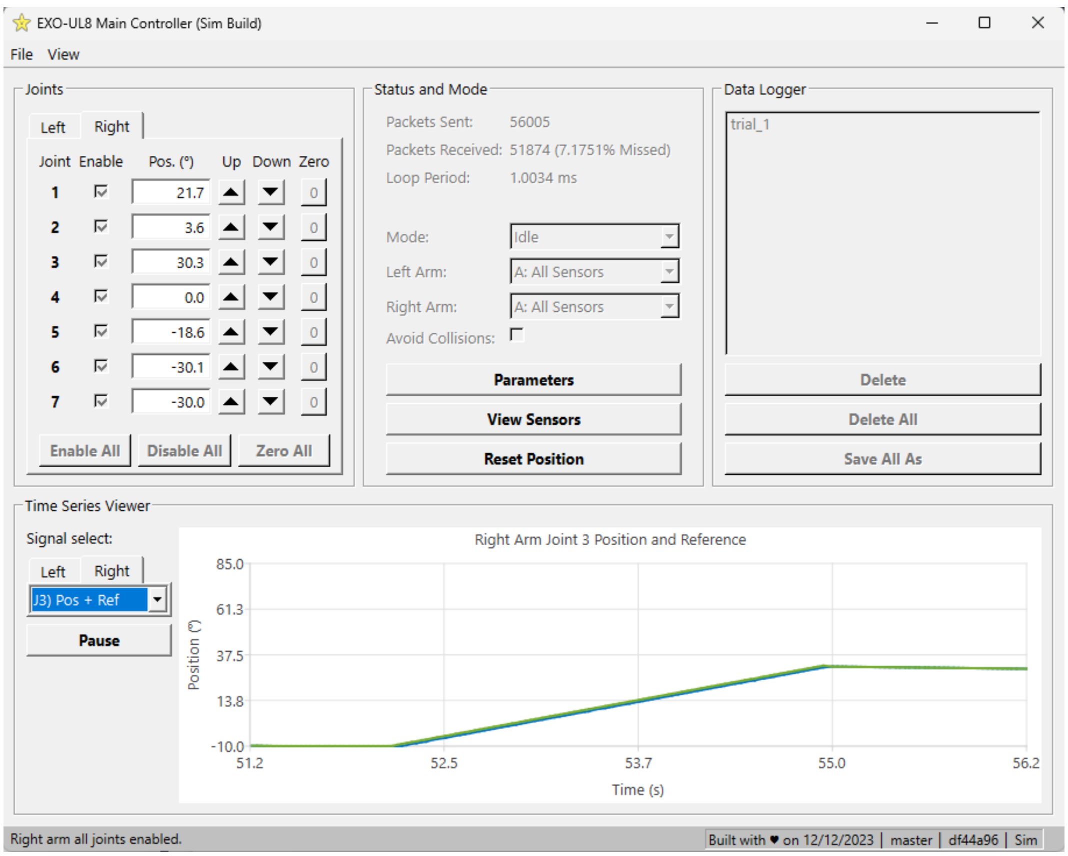


Figure 2.14: Graphical user interface of the EXO-UL8 control interface, written in C++17 and QT6.

CHAPTER 3

Admittance Control

Admittance control in pHRI is concerned with generating robot positions in response to human-applied forces. A common approach is to use a virtual dynamics model to generate the positions [4,67–70]. A measured, or estimated, human-applied force integrates the virtual dynamics, whose position and velocity can serve as reference signals to be tracked by the robot's controller. In this case, additional constraints can be imposed on the virtual state, such as ensuring that positions do not exceed certain safety regions or bounding velocity magnitudes. Since the virtual dynamics are integrated, there may be phase lag that must be considered in the robot's controller, since stability can be affected. To this end, the virtual velocity or feedforward accelerations can be used to improve the tracking performance. It should be noted that the robot will always move according to its physical dynamics. Thus, there is often a trade-off in the virtual dynamics between being lightweight (more transparent) and dynamically feasible (robot's controller can follow well).

Another technique to enable pHRI is to view the admittance controller as trying to zero the interaction force between the human and the robot. In the ideal transparent case, there is no power exchanged between the interface, since the relative velocity between the two is zero. Practically, the interaction forces are necessary for the robot's admittance controller to determine how to move. Thus, in this scheme, the robot's control is directly generated from the human-applied forces, using a stabilizing controller. While this approach may have less phase lag, it is difficult to tune or impose safety limits on position or velocity, since the robot dynamics would have to be considered in the admittance control. A comparison with the first method is further explored in [6].

The remainder of this dissertation is primarily focused on the first approach, where a virtual model generates the reference signals that a low level controller (computed torque controller for the EXO-UL8) tracks. This chapter discusses the methodology for some common virtual dynamics in section 3.1 and provides an open source C++17 library implementation: https://github.com/jianwei-sun/gtfo. Input to the virtual dynamics is also discussed; specifically, the Kalman filter-based sensor fusion technique used on the EXO-UL8 is presented.

3.1 Virtual Dynamics

The virtual model is responsible for generating reference signals corresponding to the desired motion of the pHRI system resulting from human-applied forces. Typically, the virtual system models physical systems, albeit with different parameters, so that they align well with our existing intuition for how they should react. Second-order point mass systems parameterized by mass, damping, and sometimes a virtual spring, are common. However, other systems are also worth discussing, such as first-order dynamics or second-order rigid body dynamics. Reference generation models, such as ones that generate smooth homing or point-to-point trajectories, also belong to this category.

3.1.1 Point Mass Dynamics

In the simplest case, consider a single dimensional second-order model:

$$m\ddot{p}(t) + b\dot{p}(t) = u(t), \tag{3.1}$$

where $p(\cdot) : \mathbb{R} \to \mathbb{R}$ is the position, $u(\cdot) : \mathbb{R} \to \mathbb{R}$ is the input force, $m \in \mathbb{R}_{>0}$ is the mass, and $b \in \mathbb{R}_{\geq 0}$ is the damping coefficient. Such a model is frequent in the literature since it is easy to implement and can easily be extended to the vector case. The damping parameter b often must be included to ensure passivity of the virtual system, which is necessary for stability. Although equation (3.1) is in continuous time, a discrete time implementation is possible using any discretization technique. Letting the virtual state $x := \mathrm{col}(p, \dot{p})$, the dynamics can be written in state-space format:

$$\dot{x} := \begin{bmatrix} 0 & 1 \\ 0 & -b/m \end{bmatrix} x + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u, \tag{3.2}$$

$$:= Ax + Bu \tag{3.3}$$

where the dependency on time has been dropped for clarity. Software implementation requires the dynamics to be discretized, which can be done with forward Euler:

$$x_{k+1} = A_d x_k + B_d u_k, \tag{3.4}$$

where $A_d = I + AT$, $B_d = BT$, and T is the loop period. Such a technique can cause the system to no longer be passive, so implicit Euler techniques may yield better stability:

$$\dot{p}_{k+1} = \dot{p}_k + T\left(-\frac{b}{m}\dot{p}_k + \frac{1}{m}u_k\right), \tag{3.5}$$

$$p_{k+1} = p_k + T\dot{p}_{k+1}, \tag{3.6}$$

where the updated velocity is used to compute the updated position. In some cases, it may be necessary to exactly discretize equation (3.2). To this end, assume that the input is piecewise constant and constant on each sampling period. Then, solving the differential equation over one period yields:

$$x(t+T) = e^{AT}x(t) + \int_t^{t+T} e^{A(t-\tau)}Bu(\tau)d\tau. \tag{3.7}$$

On the interval $[t, t+T)$, the input is constant: $u(\tau) = \overline{u}$. Then, simplifying the integral:

$$x(t+T) = e^{AT}x(t) + \int_t^{t+T} e^{A(t-\tau)}Bd\tau\overline{u}, \tag{3.8}$$

$$= A_d x(t) + B_d\overline{u}, \tag{3.9}$$

where:

$$A_d := e^{AT}, \tag{3.10}$$

$$B_d := \int_t^{t+T} e^{A(t-\tau)}Bd\tau. \tag{3.11}$$

In the case that A is invertible, $B_d = A^{-1}(A - I)B$.

Regardless of the discretization used (forward Euler, implicit Euler, exact), decreasing the sampling period generally yields better results, but at the cost of requiring a higher control rate.

3.1.1.1 First-Order Dynamics

First-order systems are also common and can serve as virtual dynamics. Let $x \in \mathbb{R}$ represent the virtual position, then the continuous time dynamics take the form:

$$\tau \dot{x} + x = Ku, \tag{3.12}$$

where $\tau, K \in \mathbb{R}_{>0}$ are the time-constant and gain, respectively. The dynamics can easily be exactly discretized to yield:

$$x_{k+1} = e^{-\frac{T}{\tau}} x_k + K(1 - e^{-\frac{T}{\tau}})u_k. \tag{3.13}$$

The first-order system can have less phase lag than the second-order system, but tuning the parameters for transparency can be less intuitive.

3.1.2 Rigid Body Dynamics

In general, point-mass dynamics are easy to implement since their state-space is Euclidean. However, certain applications may require more complicated dynamics, such as emulating the interaction with a virtual object. In this case, the object has geometry and cannot simply be represented by a point mass. The rigid body dynamics can be considered in two parts: the translational dynamics of its center of mass, and the rotational dynamics around the center of mass. The former can use the aforementioned second-order point-mass dynamics. However, the rotational dynamics require more care.

3.1.2.1 Rotational Dynamics

Assuming that the center of mass of the virtual object is fixed, the orientation of the object relative to a global reference frame can be expressed by an element $R \in \mathrm{SO}(3)$. Typically, R is a 3×3 rotation matrix or a unit quaternion. Local coordinates for SO(3), such as Euler angles, are often used for simplicity, but suffer from gimbal lock when the orientation leaves the coordinate chart in which the local coordinate is defined. Let $u \in \mathbb{R}^3$ be a torque acting on the body about its center of mass, expressed in the body frame. The goal is to compute the updated orientation R_{k+1} from its current orientation.

Let $\mathbb{I} \in \mathbb{R}^{3\times3}$ be the object's inertia matrix, $b \in \mathbb{R}_{\geq 0}$ be the rotational damping, and $\omega \in \mathbb{R}^3$ be the angular velocity also expressed in the body frame. Then, the dynamics follow Euler's equations with damping:

$$\mathbb{I}\dot{\omega} + \omega \times (\mathbb{I}\omega) + b\omega = u, \tag{3.14}$$

which can be discretized. The orientation then needs to be updated by a rotation $\Omega_k = \dot{\omega}_k T$. Since this rotation is about the axis $\Omega_k/\|\Omega_k\|$ by an amount $\|\Omega_k\|$, the corresponding rotation is:

$$R_T = e^{\widehat{\Omega}_k}, \tag{3.15}$$

where $\widehat{\Omega}_k \in \mathfrak{so}(3)$ is the skew-symmetric form of Ω_k, belonging to the Lie algebra of SO(3) at the identity. A closed form for the matrix exponential is given by the Rodrigues' rotation formula as:

$$e^{\widehat{\Omega}_k} = I + \frac{\widehat{\Omega}_k}{\|\Omega_k\|}\sin(\|\Omega_k\|) + \left(\frac{\widehat{\Omega}_k}{\|\Omega_k\|}\right)^2 (1 - \cos(\|\Omega_k\|)). \tag{3.16}$$

In the case that $\|\Omega_k\| \approx 0$, small angle approximations or Taylor expansions should be used so that floating-point division by $\|\Omega_k\|$ does not cause instability. Once the delta rotation is computed, the orientation can be updated as:

$$R_{k+1} = R_k R_T. \tag{3.17}$$

In the case that unit quaternions are used to represent the orientation, the delta quaternion in coordinates $q_T = (w, x, y, z)$ can be computed with:

$$q_T = (\cos(\theta), s\Omega_k) \tag{3.18}$$

where $\theta = (T/2)\|\Omega_k\|$, $s = (T/2)\,\mathrm{sinc}(\theta)$, and $\mathrm{sinc} : \mathbb{R} \to \mathbb{R}$ is the cardinal sine function defined as:

$$\mathrm{sinc}(x) := \begin{cases} \frac{\sin x}{x}, & x \neq 0, \\ 1, & x = 0. \end{cases} \tag{3.19}$$

Numerical implementation of sinc can make the approximation:

$$\mathrm{sinc}(x) \approx \begin{cases} 1 - \frac{x^2}{6}, & \|x\| < \epsilon, \\ \frac{\sin x}{x}, & \|x\| >= \epsilon, \end{cases} \tag{3.20}$$

for some numerical tolerance $\epsilon \in \mathbb{R}_{>0}$. As long as a globally valid representation is used for orientation, e.g. rotation matrices, unit quaternions, axis-angles, the orientation dynamics do not suffer from gimbal lock.

3.1.2.2 Rigid Body Dynamics

Given the rotation and point mass dynamics, the rigid body dynamics can be viewed as dynamics on $\mathbb{R}^3 \times \mathrm{SO}(3)$, where the center of mass follows the second-order point mass dynamics, and the orientation follows the rotational dynamics. Typically, the human-applied input is measured by a wrench sensor located at the end effector of the robot. In this case, it should be noted that the wrench measurement is in the body frame of the virtual object. When the point mass dynamics representing the object's center of mass are used, the force component of the human-applied wrench needs to be rotated into the global frame: $f_g = R_{gb} f_b$, where R is the body frame relative to the global frame. The torque component is fine to leave in the body frame, which is typically done in practice.

An open source C++17 templated virtual dynamics library is implemented to facilitate pHRI studies and experiments for the Bionics Lab: https://github.com/jianwei-sun/gtfo.

3.2 EXO-UL8 Sensor Fusion

For systems that have multiple force/torque sensors, such as the EXO-UL8, the wrench measurements must be combined before they can propagate the virtual model in the admittance controller. One method for fusing the measurements is with a Kalman filter [6]. Using the EXO-UL8 as an example, this section discusses the implementation for fusing the three wrenches measured from the upper, lower, and wrist sensors into a joint-space torque command used to propagate the virtual dynamics.

Each of the three measured wrenches is transformed into the global frame and then mapped to joint torques using the spatial manipulator Jacobian. The resulting joint torques $(\Gamma_u, \Gamma_l, \Gamma_w)$ are then combined to propagate the virtual dynamics.

A Kalman filter is utilized for the sensor fusion system. The Kalman filter-based sensor fusion combines the torques from the sensors $(\Gamma_u, \Gamma_l, \Gamma_w)$ into a single torque estimate $\hat{\Gamma}$. Since the joint torques are generated from human-applied forces, the exact signal is not known a priori. Therefore, the process equation for Γ is modeled as a random walk, similar to the technique used in [6, 70]:

$$\Gamma[k+1] = \Gamma[k] + (T)w_\Gamma[k], \tag{3.21}$$

where T is the sampling period, and $w_\Gamma[k] \sim \mathcal{N}(0, Q_\Gamma)$, where Q_Γ is an empirically tuned covariance matrix. The torques $\Gamma_u, \Gamma_l, \Gamma_w$ are then treated as measurements with additive Gaussian noise to the Kalman filter:

$$z[k] := \begin{bmatrix} \Gamma_u[k] \\ \Gamma_l[k] \\ \Gamma_w[k] \end{bmatrix} + \begin{bmatrix} w_u[k] \\ w_l[k] \\ w_w[k] \end{bmatrix}, \tag{3.22}$$

$$= \begin{bmatrix} \mathbb{I}_{3\times 3} & \mathbb{0}_{3\times 4} \\ \mathbb{I}_{5\times 5} & \mathbb{0}_{5\times 2} \\ \multicolumn{2}{c}{\mathbb{I}_{7\times 7}} \end{bmatrix} \Gamma[k] + \begin{bmatrix} w_u[k] \\ w_l[k] \\ w_w[k] \end{bmatrix}, \tag{3.23}$$

$$:= H\Gamma[k] + \mathrm{col}(w_u[k], w_l[k], w_w[k]), \tag{3.24}$$

where $z[k] \in \mathbb{R}^{15}$ is a combined vector of joint torques from the sensors. $w_u[k] \sim \mathcal{N}(\mathbb{0}_{3\times 1}, R_u)$, $w_l[k] \sim \mathcal{N}(\mathbb{0}_{5\times 1}, R_l)$, and $w_w[k] \sim \mathcal{N}(\mathbb{0}_{7\times 1}, R_w)$, where $R_u \in \mathbb{R}^{3\times 3}$, $R_l \in \mathbb{R}^{5\times 5}$, and $R_w \in \mathbb{R}^{7\times 7}$ are the noise covariance matrices corresponding to the upper, lower, and wrist sensor, respectively. Let $\hat{\Gamma} \in \mathbb{R}^7$ be the *minimum mean squared error* (MMSE) estimate of Γ, $P_p \in \mathbb{R}^{7\times 7}$ be the variance of the a priori, $P_m \in \mathbb{R}^{7\times 7}$ be the variance of the a posteriori, and $R := \mathrm{diag}(R_u, R_l, R_w)$. Then, the update equations for the Kalman filter become:

Initialization:

$$\hat{\Gamma}[0] = \mathbb{0}_{7\times 1}, \tag{3.25}$$

$$P_m[0] = (T)^2 Q_\Gamma. \tag{3.26}$$

A Priori Update:

$$P_p[k] = P_m[k-1] + (T)^2 Q_\Gamma. \tag{3.27}$$

A Posteriori Update:

$$K[k] := P_p[k]H^\top (HP_p[k]H^\top + R)^{-1}, \tag{3.28}$$

$$\hat{\Gamma}[k] = (\mathbb{I} - K[k]H)\hat{\Gamma}[k-1] + K[k]z[k], \tag{3.29}$$

$$P_m[k] = (\mathbb{I} - K[k]H)P_p[k](\mathbb{I} - K[k]H)^\top + K[k]RK[k]^\top, \tag{3.30}$$

where $K[k] \in \mathbb{R}^{7\times 15}$ is defined as the Kalman gain at time step k. Note that equation (3.30) implements the Joseph form for numerical stability.

Convergence of the Kalman filter is guaranteed by the pair $(\mathbb{I}_{7\times 7}, H)$ being detectable and the pair $(\mathbb{I}_{7\times 7}, Q_\Gamma^{1/2})$ being stabilizable [71], where $\mathbb{I}_{7\times 7}$ is the state transition matrix in equation (3.21). Then, let P_∞ be the steady-state a posteriori variance calculated from the discrete algebraic Riccati equation and let $K_\infty = P_\infty H^\top (H P_\infty H^\top + R)^{-1}$ be the steady-state Kalman gain [72]. The converged update equations become:

$$\hat{\Gamma}[k] = (\mathbb{I} - K_\infty H)\hat{\Gamma}[k-1] + K_\infty z[k], \tag{3.31}$$

which is a discrete-time, linear time-invariant system.

CHAPTER 4

Reference-Level Safety

[2] **J. Sun**, Erik Harrison Kramer, J. Rosen, "A Safety-Focused Admittance Control Approach for Physical Human-Robot Interaction with Rigid Multi-Arm Serial Link Exoskeletons." *Under review.*

4.1 Overview

Ensuring safety in physical human-robot interaction is challenging due to hardware and control architecture differences across robots, and is often implemented as system-dependent ad-hoc approaches. To offer a holistic solution, we present a hardware-independent safety-focused admittance control approach which promotes safety at the reference-generation level. This safety framework can restrict virtual dynamics through soft virtual bounds. Hard bounds are also introduced as a way to impose infinitely stiff soft bounds. As part of the overall approach, we also present a method for serial manipulator and multi-segment entity collision avoidance by using partial Jacobians. In order to demonstrate the methodology's versatility across hardware platforms, we experimentally validate on two robotic systems: (1) the V-Rex, a non-anthropomorphic full-body haptic device composed of five robotic arms interacting with the body at the hands, feet, and pelvis; and (2) the EXO-UL8, an anthropomorphic bimanual upper-limb exoskeleton; which exist on opposite ends of the task/joint space control, non-redundant/redundant, off-the-shelf (industrial)/custom, non-anthropomorphic/anthropomorphic spectra. Experimental results validate virtual dynamics, soft and hard bounds, and multi-arm collision avoidance on both systems. In all cases, both systems respect bound and collision constraints, supporting the approach as a safety-focused admittance control design.

4.2 Introduction

In pHRI, admittance control is often used to generate desired trajectories for controlling how a robot should respond to human-applied forces. In any human-interaction, safety is paramount, and must therefore be addressed at not only the hardware level, but also throughout the control. This paper presents a safety-focused admittance control approach for pHRI with rigid multi-arm serial link exoskeletons.

Safety in pHRI is a complex problem that has no perfect solution, but has been approached through many ways. Most directly, mechanical safety, such as joint-stops [57, 73], back-driveable actuators [57,74], e-stops [57,75], and compliance [76,77], provide limited degrees of safety. While actuator output can be reduced to levels that minimize harm, doing so sacrifices performance and can result in undesirable behavior [78]. E-stops require human input, which may be delayed during dangerous situations. Compliance and soft interfaces may be mechanically simple to implement, but can significantly complicate the control [76, 77]. Furthermore, these mechanical solutions may not be viable to off-the-shelf manipulators that were not designed with pHRI in mind, and are better suited for custom designs.

Safety research also includes controller-level approaches, which can be implemented entirely in software and/or rely on minor hardware additions. These include: human-tracking through vision [79] or motion-based sensors [80]; monitoring instability through indices or heuristics [67,81]; saturating or filtering some aspect of human-applied signals [4,82,83]; and so on. While human tracking methods can provide useful real-time information, they may require additional complexity such as more sensors [79,80], and/or placing sensors directly on the human [42, 48], which detracts from ease of use. Detection of unsafe interaction through performance metrics or heuristics has shown good results [67, 81], but are usually specific to the system and difficult to generalize. Saturation of signals is not necessarily safer because the system can be slower to respond to unsafe situations.

Safety can also be tackled at the reference-generation level, which includes: dynamically tuning admittance control parameters [67–69, 84], defining virtual bounding regions [85–89], utilizing data-driven methods [69, 88], and collision avoidance [90–97]. Collision avoidance involves two aspects: detection (finding intersections between 3D shapes representing the robot's links) and avoidance (using robot kinematics to prevent dangerous motions). Existing simulation environments, such as MuJoCo [66], implement detection by finding intersections between convex hulls of the links. Although most accurate, the computation cost may not be justified in pHRI, which would want large bounding regions around human-operated exoskeletons. Other approaches make this trade-off by using simpler geometries [92,94,95]. [92] implements avoidance in task-space, requiring the Jacobian to be invertible which poses a problem for redundant manipulators. In [94] and [95], a series of virtual spheres encapsulate the manipulator, and the velocity between colliding spheres is restricted. However, adjusting the detection radius would modify their number and positions, requiring Jacobian recomputation for each sphere. [96] and [97] restrict the relative velocity between closest points on colliding manipulators. Whereas [96] assumes the closest points can be determined and only restricts the velocity to a fixed value, [97] uses an iterative scheme to compute these points and a quadratic program to restrict the velocity.

Our approach improves upon previous methods and extends the collision avoidance methodology of [92] by using line segments to provide analytic expressions for distances between primitives and utilizing the method from [98] to analytically compute partial Jacobians at any arbitrary collision point.

In this paper, we present a comprehensive safety-oriented admittance control framework for pHRI with rigid multi-arm serial link exoskeletons. We validate this approach on two multi-arm exoskeletons which have significantly different hardware and control. Our approach facilitates admittance control by generating collision-free trajectories that emulate virtual second-order systems, while incorporating hard and soft boundary constraints for operation within a safe workspace.

Within this unified safety framework, we introduce a new method for emulating infinitely stiff virtual bounds (sometimes referred to as virtual fixtures in robotic surgery [99]), which avoids issues of stability typically seen with high-stiffness virtual springs. We also present a method for collision avoidance between multi-segment entities by utilizing partial Jacobians. This collision avoidance technique uses the translational components of a robot manipulator Jacobian evaluated at potential collision points to determine safe motion directions, preventing collisions with not only the end effector, but also any point along the manipulator's links.

In summary, our contribution is a broad safety-centric admittance control approach that is comprised of emulating infinitely stiff virtual bounds and avoiding collisions for serial link manipulators. This framework can serve as both a safe self-contained reference generator for admittance control and a versatile lightweight safety intermediary layer, upon which customized applications can be developed. Moreover, individual elements of this framework can be employed modularly to enhance specific safety functionalities of pHRI systems.

4.3 Methodology

To enable pHRI, admittance control is used to generate motions of a virtual system with some desired dynamics from physical human-applied forces. The robot's controller then tracks these target trajectories, making the robot a physical manifestation of the virtual system. This behavior forms the foundation for advanced features such as safety bounds (limiting movement and speed) and multi-arm collision avoidance.

4.3.1 Virtual Dynamics

Emulating how a light-weight system moves is commonly done by propagating a virtual second-order mass-damper model [4, 67–69]. The second-order dynamics are chosen due to their similarities with physical mechanical systems and our existing intuition about how they react. Furthermore, their parameters are physically intuitive and can be easily tuned. The dynamics in a single dimension are:

$$m\ddot{p}(t) + b\dot{p}(t) = f(t), \tag{4.1}$$

where $p(\cdot) : \mathbb{R} \to \mathbb{R}$ is the position, $f(\cdot) : \mathbb{R} \to \mathbb{R}$ is the input force, $m \in \mathbb{R}_{>0}$ is the mass, and $b \in \mathbb{R}_{\geq 0}$ is the damping coefficient. The dynamics of vector systems are similar, in that each coordinate implements the scalar dynamics, allowing the system to be used for both task or joint-space admittance control. The state variables' explicit dependency on time will no longer be shown for brevity.

4.3.2 Bounds

One approach for safety in pHRI is to define regions in which motions are hindered. To this end, we define a bound, $B \subset \mathbb{R}^n$, as a convex closed subset of the space of virtual positions, where n is the dimension of the virtual system. Convexity is necessary so that projections into B, denoted by $\mathrm{proj}_B(\cdot) : \mathbb{R}^n \to B$, are unique. From practice, we saw that norm bounds and rectangular bounds are most commonly used, so we focus on them. However, the methodology applies to any convex bound.

Moreover, we classify bounds as soft or hard. Soft bounds allow violation but generate a restoring force as a function of how far the virtual position is beyond the bound (typically implemented as a virtual spring-damper system), similar to impedance-based schemes [87, 100] or virtual fixtures [99, 101]. Stiff or rigid boundaries can be simulated using a high stiffness constant for the virtual spring, but such approaches can cause stability issues [99].

To address the limitation of using soft bounds to emulate infinitely stiff boundaries, we introduce hard bounds, which are inviolable at the reference generation level. These bounds constrain the virtual position and velocity directly, avoiding the need to integrate large restoring forces. Hard bounds are particularly useful for defining mechanical endpoints, such as joint limits or physical boundaries within task space. These bounds represent the safe regions in which the pHRI should occur. Thus, the virtual position must always start within the bound for the method to be valid. We base this on the fact that the human should not even be interacting with the robot if it is outside safety limits. Using hard bounds alone can lead to abrupt velocity changes at their boundaries, which may be uncomfortable for an operator moving rapidly towards them. Therefore, our approach is designed to allow utilization of both types of bounds simultaneously, with the soft bound contained within the hard bound. This formulation allows for soft bounds to buffer motions towards a hard bound, meaning fast motions are gradually slowed instead of abruptly stopped.

4.3.2.1 Soft Bounds

Let $p_k \in \mathbb{R}^n, v_k \in \mathbb{R}^n$ be the position and velocity of the virtual system at the current k^{th} timestep, respectively. Furthermore, let $B \subset \mathbb{R}^n$ be a bound. To emulate a restoring force based on how much B is violated, let $r \in \mathbb{R}^n$ be the vector from the current position to the closest bounded position:

$$r := p_k - \text{proj}_B(p_k), \tag{4.2}$$

which is well-defined and unique due to the convexity of B. Then, when r is nonzero, let $\hat{r} := \|r\|^{-1}r$, and the restoring force $f_r \in \mathbb{R}^n$ can emulate a virtual spring-damper system:

$$f_r = -k_s r - \max\{v_k \cdot \hat{r}, 0\} d_s \hat{r}, \tag{4.3}$$

where $k_s \in \mathbb{R}_{\geq 0}$ and $d_s \in \mathbb{R}_{\geq 0}$ are the restoring spring constant and damping coefficient, respectively. Note that all vector norms are the 2-norm, unless otherwise specified. The max function is included so that the restoring force only hinders movement that further violates the bound. The restoring force is then added to equation (4.1) before the dynamics are propagated.

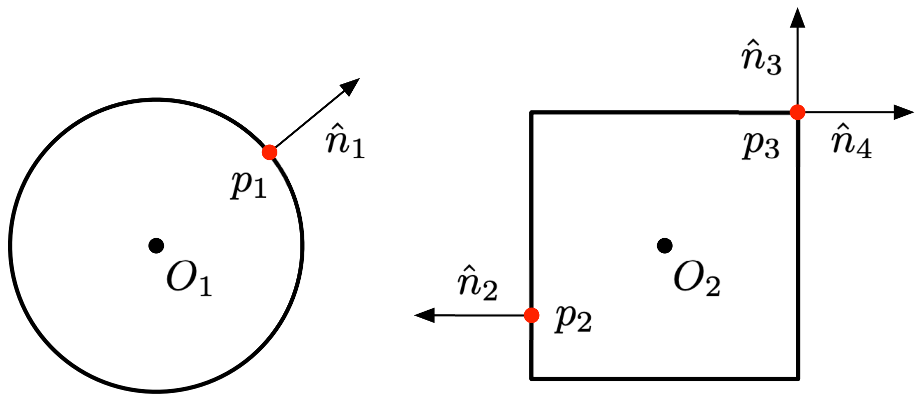


Figure 4.1: Convex bounds, such as norms and rectangles, can be used as both soft and hard bounds. In the norm bound, there is only ever a single unit surface normal, so $S_{p_1} = \{\hat{n}_1\}$. However, the rectangular bound is not differentiable at the corners, so $S_{p_2} = \{\hat{n}_2\}$ and $S_{p_3} = \{\hat{n}_3, \hat{n}_4\}$.

4.3.2.2 Hard Bounds

Similar to soft bounds, a hard bound B is also a convex closed subset of the space of virtual positions. However, to ensure that p_k is always contained within B, first assume that the unbounded discretized dynamics of equation (4.1) would result in $p_{k+1} \notin B$. Then, the virtual position can be updated as $p'_{k+1} := \text{proj}_B(p_{k+1})$, which would ensure containment within B.

For velocities, the components that try to escape B should be removed. Let $s(\cdot) : \partial B \to \mathbb{R}^n$ be a map from points on the surface of B to its corresponding unit normal vector. Since $s(\cdot)$ may not be defined everywhere on ∂B, e.g., at the corners if B is a rectangle, define the unit surface normal at such a point p as a set, S_p, of all possible limits of $s(\cdot)$ towards p. For bounds where $s(\cdot)$ exists everywhere on ∂B, $|S_p| = 1$. For other bounds, such as the rectangle example, there exist p such that $|S_p| \geq 1$, as shown in Fig. 4.1.

To ensure that future states of the virtual dynamics remain bounded, update the position and velocity as follows:

$$\tilde{p}_{k+1} = \text{proj}_B(p_{k+1}), \tag{4.4}$$

$$\begin{aligned} \tilde{v}_{k+1} = \underset{v}{\text{argmin}} \quad & \|v - v_{k+1}\|^2 \\ \text{s.t.} \quad & \hat{n}^\top v \leq 0,\ \forall \hat{n} \in S_{\tilde{p}_{k+1}} \end{aligned} \tag{4.5}$$

Positions are guaranteed to always remain in B. The update in velocity ensures that components that try to escape the bound are zeroed, which is necessary for robot systems that only use the virtual velocity as reference. When the bound is rectangular such that all the unit normal vectors in S_p are orthogonal, then equation (4.5) can be simplified to:

$$\tilde{v}_{k+1} = v_{k+1} - \sum_{\hat{n} \in S_{\tilde{p}_{k+1}}} \max\{v_{k+1} \cdot \hat{n}, 0\}\hat{n}. \tag{4.6}$$

On each integration step, soft bounds are evaluated before virtual dynamics are propagated with equation (4.1) in order to integrate the restoring force. Then, hard bounds are enforced with equations (4.4) and (4.5) to ensure they are always satisfied. The same hard bound methodology in equations (4.4) and (4.5) can be imposed on any higher-order derivative of position. This allows for further constraints on virtual quantities such as velocity or acceleration.

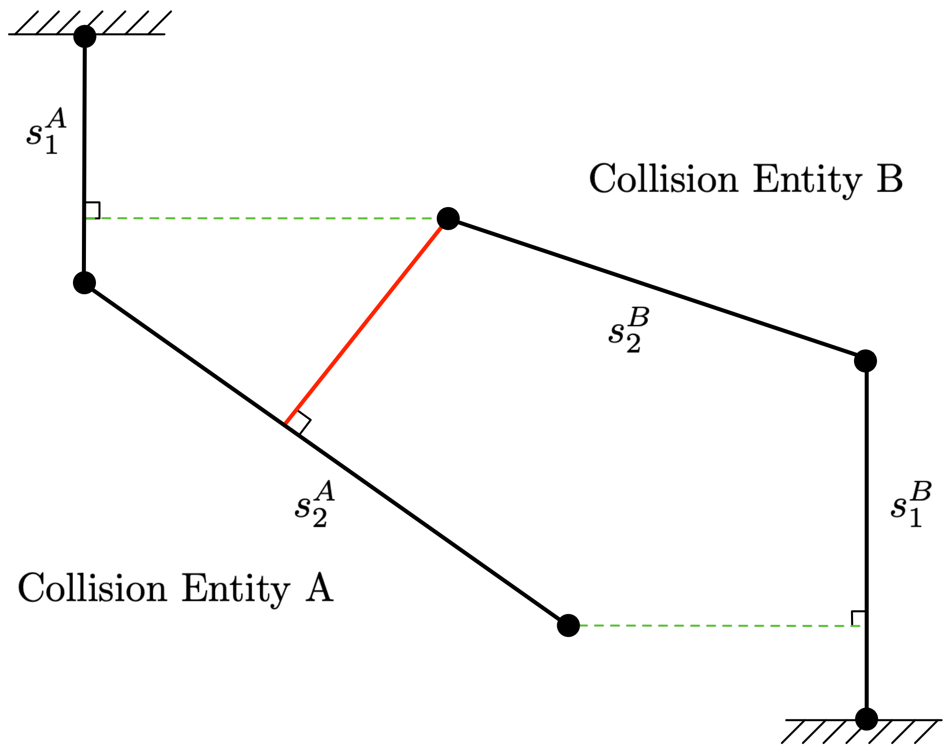


Figure 4.2: A sample collision shown between two entities, each of which is represented by three vertices and two corresponding segments: $E_A = \{s_1^A, s_2^A\}$ and $E_B = \{s_1^B, s_2^B\}$. Green dotted lines indicate potential collisions that do not exceed the threshold ϵ, whereas the red line shows the collision. The intersections of the red line with s_2^A and s_2^B constitute the collided sets $\tilde{\mathcal{A}}_A$ and $\tilde{\mathcal{A}}_B$.

4.3.3 Collisions

Collision avoidance is an integral part of safety in pHRI, especially for multi-arm systems. Any controllable object with which a collision can occur is henceforth denoted a controllable entity. Objects that cannot be controlled, such as the human, obstacles, or static links and frames of a robot, are called uncontrollable entities. Uncontrollable entities can be free or fixed. As mentioned in the literature review, collision avoidance consists of two aspects: detection and avoidance. Detection is concerned with finding entities that are too close, whereas avoidance then restricts the relative motion between the entities from becoming closer.

4.3.3.1 Detection

Let $V_j^i \in \mathbb{R}^3$, $j \in \{1, \ldots, m_i\}$, $m_i \geq 1$ be called entity vertices, which are a series of user-specified locations along entity i in the system, such that the segment drawn between adjacent points captures the geometry of the corresponding link of the entity, as shown in Fig. 4.2. m_i is the number of entity vertices on entity i. Let $s_j^i := \{\lambda V_j^i + (1-\lambda)V_{j+1}^i \mid \lambda \in [0,1] \subset \mathbb{R}\}$ be the segment between two adjacent collision points, and $\mathcal{S}$ be the space of segments. Then, let $E_i := \{s_j^i \mid j \in \{1, \ldots, m_i\}\}$ be the set of the aforementioned segments corresponding to entity i. If $m_i = 1$, then $E_i = \{\{V_1^i\}\}$ consists of a single zero-length segment.

Next, for each entity, enumerate the set $\mathcal{C}_i = \{(s_a, s_b) \mid s_a \in E_i, s_b \notin E_i\}$, which is the set of potential collision segments between entity i and every other entity that has collision points. Certain collisions, such as self-collisions between adjacent links or collisions between robots that are physically farther than the limits of their workspace, can be pruned from this set in order to speed up computation. Let the pruned set be $\tilde{\mathcal{C}}_i \subseteq \mathcal{C}_i$.

Let $d(\cdot) : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^3 \times \mathbb{R}^3$ map segments s_a, s_b to points $c_a \in s_a, c_b \in s_b$ such that $\|c_a - c_b\|$ is minimized. To make $d(\cdot)$ well-defined, when s_a and s_b are parallel and proximate, c_a and c_b are chosen as the midpoints of the overlapping region. The definition for $d(\cdot)$ is discussed below. For entity i, let $\mathcal{A}_i := \{d(s_a, s_b) \mid (s_a, s_b) \in \tilde{\mathcal{C}}_i\}$ and compute $\tilde{\mathcal{A}}_i := \{(c_a, c_b) \mid (c_a, c_b) \in \mathcal{A}_i, \|c_a - c_b\| \leq \epsilon_i\}$, called the collided set, where $\epsilon_i \in \mathbb{R}_{\geq 0}$ is a user-specified threshold for detection. Note that $\tilde{\mathcal{A}}_i$ can be computed in parallel. The collided set tells not only whether a collision has occurred, but also its location and direction.

The aforementioned segment-segment distance function $d(\cdot)$ needs to consider corner cases, such as: zero-length segments and parallel-segments. Let two segments $s_a, s_b \in \mathcal{S}$ have endpoints $a_0, a_1 \in \mathbb{R}^3$ and $b_0, b_1 \in \mathbb{R}^3$, respectively.

If $a_0 = a_1$ and $b_0 = b_1$, then $d(s_a, s_b) = (a_0, b_0)$.

If only one of the segments has zero-length, say $a_0 = a_1$ without loss of generality, then project a_0 onto s_b and determine its ratio $r_b \in \mathbb{R}$ along s_b:

$$r_b = \frac{(b_1 - b_0)^\top (a_0 - b_0)}{\|b_1 - b_0\|^2}. \tag{4.7}$$

Then, depending on the value of r_b, assign the point on s_b:

$$d(s_a, s_b) = \begin{cases} (a_0, b_0), & r_b < 0, \\ (a_0, b_0 + r_b(b_1 - b_0)), & 0 \leq r_b \leq 1, \\ (a_0, b_1), & 1 < r_b. \end{cases} \tag{4.8}$$

If both segments have nonzero length, then check if they are parallel. If so, i.e., $(a_1 - a_0) \times (b_1 - b_0) = 0$, then find the midpoints of the overlapping region: $a_m \in s_a$ and $b_m \in s_b$. Then, $d(s_a, s_b) = (a_m, b_m)$. Otherwise, compute the mutual-perpendicular segment s_p with endpoints $p_a \in s_a$ and $p_b \in s_b$. This can be done by finding the ratios of p_a along s_a and p_b along s_b.Now, define the normalized cross-product and compute the ratio r_a along segment s_a as:

$$\hat{c} := \frac{a_1 - a_0}{\|a_1 - a_0\|} \times \frac{b_1 - b_0}{\|b_1 - b_0\|}, \tag{4.9}$$

$$r_a = \frac{1}{\|a_1 - a_0\| \|\hat{c}\|^2} \det \left(\begin{bmatrix} (b_0 - a_0)^\top \\ \frac{(b_1 - b_0)^\top}{\|b_1 - b_0\|} \\ \hat{c}^\top \end{bmatrix} \right), \tag{4.10}$$

and similarly for r_b. Then, $d(s_a, s_b) = (a_0 + r_a(a_1 - a_0), b_0 + r_b(b_1 - b_0))$. Note that this computation also handles the case when s_a and s_b intersect.

4.3.3.2 Avoidance

Collision avoidance can only be implemented on controllable entities, which are typically robot manipulators. It is implemented in joint-space in order to also consider internal motions of redundant manipulators, which is particularly relevant for exoskeletons. The objective is to restrict motions of the manipulator such that for each $(c_a, c_b) \in \tilde{\mathcal{A}}_i$, the point c_a is prohibited from moving in the direction $c_b - c_a$. Since c_a can be anywhere on the manipulator, the kinematics of the robot are required. However, to be robot-agnostic, our approach requires the translational components of the Jacobian functions of the robot on which it is implemented: $J_j^i(\cdot) : \mathbb{R}^3 \times \mathbb{R}^n \to \mathbb{R}^{3\times n}$, where n is the DoFs of the robot, henceforth referred to as the partial Jacobian. The output of $J_j^i(\cdot)$ is the upper three rows (linear motion) of the spatial manipulator Jacobian for an arbitrary point c_a located on segment s_j^i. A partial Jacobian can be constructed online using [98, equation (14)].

Given a desired joint-space velocity $\dot{\theta}_d \in \mathbb{R}^n$, either from the virtual dynamics if they are configured as being in joint-space, or from the robot's inverse kinematics if the virtual dynamics are in task-space, the objective is to restrict the reference joint-space velocity such that the velocity of each collided point c_a does not have positive dot product with $c_b - c_a$. More succinctly, the restricted desired joint-space velocity $\dot{\theta}_r \in \mathbb{R}^n$ can be found by solving the following quadratic program:

$$\begin{aligned} \min_{\dot{\theta}_r} \quad & \|\dot{\theta}_r - \dot{\theta}_d\|^2 \\ \text{s.t.} \quad & (c_b - c_a)^\top J_j^i(c_a, \theta)\dot{\theta}_r \leq 0 \\ & \forall (c_a, c_b) \in \tilde{\mathcal{A}}_i \end{aligned} \tag{4.11}$$

where θ is the physical joint configuration, and each J_j^i is picked such that c_a corresponds to segment s_j^i. Note that each collided point imposes one additional linear constraint on $\dot{\theta}_r$. The optimal $\dot{\theta}_r$ is then sent to the manipulator's controller and also used to update the velocity of the virtual model.

The proposed methodology of constraining potential collision points from moving closer to each other is not inherently restricted to line segments, and can be extended to other geometries in which a similar function to $d(\cdot)$ - which computes the closest points between two objects - can be constructed. For instance, planes can also be used to represent static obstacles such as the floor or walls.

4.3.4 Summary

Algorithm 1 outlines the approach, which runs within the system's control loop.

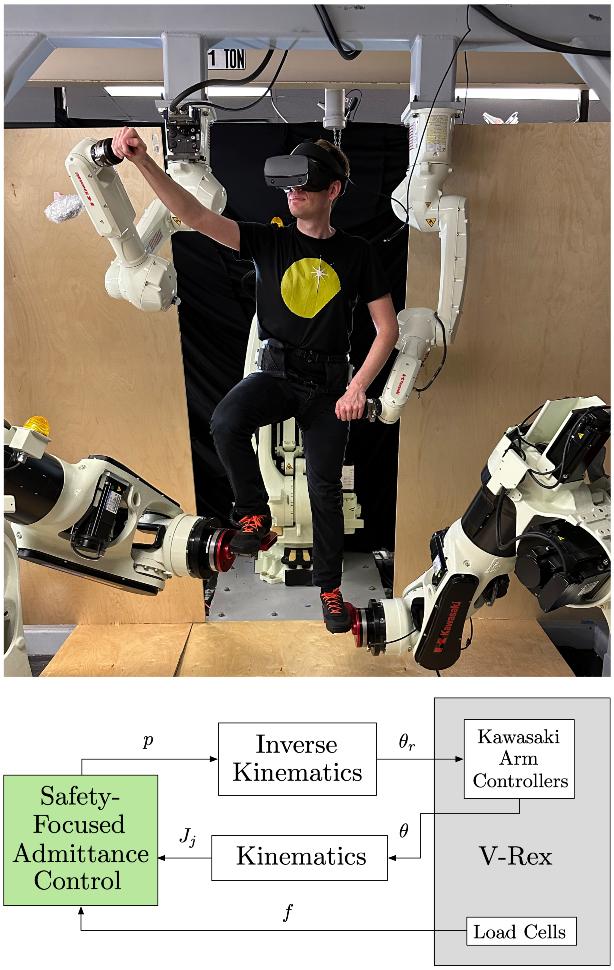
Procedure 1 Safety-Focused Admittance Control

for each controllable entity, E_i
    Receive human-applied force measurements f_h
    Compute restoring Soft Bound force f_r (equation 4.3)
    Sum dynamics input force $f = f_h + f_r$
    Propagate virtual dynamics (equation 4.1)
    Restrict state via Hard Bounds (equations 4.4, 4.5)
    Update entity vertices using physical position
    Compute collided set $\tilde{\mathcal{A}}_i$
    Restrict virtual velocity (equation 4.11)
    Output virtual state to E_i's controller

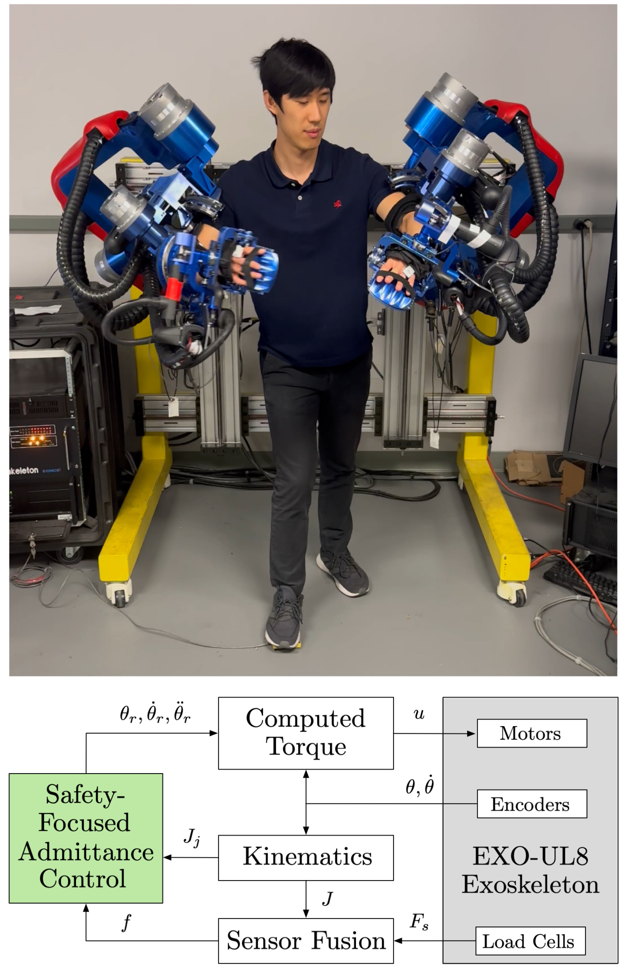
4.4 Experiment Setup

To demonstrate the correctness and versatility of the approach, a series of experiments on two different multi-arm robotic systems: (1) the V-Rex and (2) the EXO-UL8, are conducted. The V-Rex is a non-anthropomorphic exoskeleton consisting of multiple task-space-controlled off-the-shelf industrial serial manipulators, whereas the EXO-UL8 is a custom-built anthropomorphic bimanual upper-limb exoskeleton controlled in joint-space. The existence of the two systems on opposite ends of the off-the-shelf (industrial) vs. custom, task-space vs. joint-space, and non-anthropomorphic/anthropomorphic spectra demonstrates the versatility and generalizability of the approach. The experiments verify and validate both soft and hard bounds, as well as multi-arm collision avoidance, on each of the two systems.

4.4.1 Systems



(a) The V-Rex is a full body haptic device consisting of five robotic arms, designed for pHRI with virtual environments. Force-torque input from load cells is used to step virtual dynamics and determine the next position reference, p in task space. Physical joint state, θ, is read and utilized to find partial Jacobians, J_j, to adjust reference position for collision avoidance. Utilizing an inverse kinematic solver, all viable joint space configurations are found. A l^2-norm is then applied to find the closest solution to the current joint state, θ. The selected joint state is vetted through a secondary safety check before the joint references θ_r are updated for the Kawasaki arm controller for precise position control. The entire control cycle runs at 500Hz.



(b) The EXO-UL8 is a bimanual upper-limb exoskeleton designed for pHRI and robot-assisted rehabilitation. Human-applied forces are measured by three load cells on each arm located at the: upper-arm (u), lower-arm (l), and wrist (w). The measured wrenches, F_s, $s \in \{u, l, w\}$, are transformed into joint torques through the Jacobian, J, and then fused together into a single joint-space human-applied torque vector, f, with the algorithm in [5]. This torque propagates the virtual dynamics in the admittance control, which also uses the partial Jacobians, J_j for collision avoidance. The virtual states are then tracked by the exoskeleton's computed torque controller, which outputs motor torques, u. The control rate is 1KHz.

Figure 4.3: The V-Rex and EXO-UL8 systems exist on opposite ends of the task/joint space control, non-redundant/redundant, off-the-shelf (industrial)/custom, and non-anthropomorphic/anthropomorphic spectra.

4.4.1.1 V-Rex

The Virtual Reality Exoskeleton (V-Rex) is a full body haptic device designed to provide force feedback to a human interacting with a virtual environment visualized with a VR headset. The system is composed of five task-space-controlled off-the-shelf Kawasaki industrial serial manipulator robots. Two RS-007L robots are gripped by the operator's hands; two BX-100S robots are connected with breakaways to the operator's shoes; and one CX-210L provides gravity offloading through a flying harness. All robot arms have load cells between their end effectors and the human interface (three ATI Omega sensors for the lower limbs and body support plus two ATI Gamma sensors for the upper limbs). Each manipulator takes human-applied wrenches and propagates them into task-space motions of the end effector using the model shown in Fig. 4.3a.

4.4.1.2 EXO-UL8

The EXO-UL8 is a custom bimanual redundant powered joint-space controlled upper-limb anthropomorphic exoskeleton with seven DoFs in each arm, built as the latest system in a series of exoskeletons designed for pHRI and robot-assisted rehabilitation [4–6, 57–61, 102]. The HITL requirement has motivated specific hardware designs, such as rotating ring joints to allow shoulder rotation and forearm supination/pronation [57], and anatomically similar joint limits to align the exoskeleton's and operator's arms. pHRI is enabled by joint-space admittance control. Wrenches measured from three ATI Mini40 sensors on each arm are fused and propagate virtual dynamics [5], whose trajectories are tracked by a computed-torque controller [38, 40], as shown in Fig. 4.3b.

4.4.2 Experiments

4.4.2.1 Bounds

Hard and soft bounds are first demonstrated separately on the V-Rex. A single arm of the system is restricted to move in a 2D plane parallel with the ground, as shown in Fig. 4.4. For both bound types, the operator tries to move the end effector to follow the desired trajectory at a constant speed. The operator starts in the region's center, moves to the bottom-left side, traverses the desired trajectory in a counter-clockwise direction, and then returns to the center. In the hard bounds trial, a square hard bound of equal size (side length: 37 cm) to the trajectory rotated by 45° is imposed. The soft bound trial utilizes the same bound shape, but has soft bound parameters: $k_s = 250$ N/m, $d_s = 60$ Ns/m. The manipulator implements the same second-order virtual dynamics parameters $m = 10$ kg, $d = 15$ Ns/m for both trials.

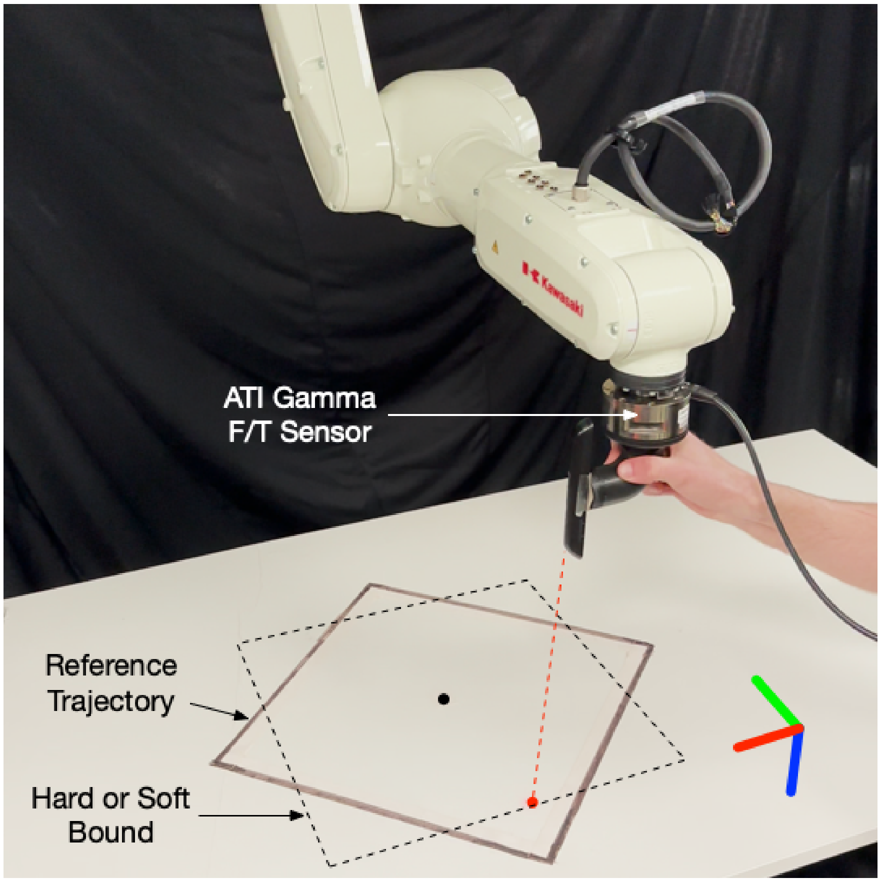


Figure 4.4: For bounds experiments on the V-Rex, the operator is guided by a black square (solid line) on the surface beneath one of upper limb arms. A square boundary (shown by the dashed line), either soft or hard, is positioned at a 45° angle to the trajectory, sharing the same centroid. Starting from the centroid, the operator follows the trajectory to the best of their ability before returning to the starting point. A laser pointer attached to the end effector helps the operator track their progress throughout the experiment.

Since the EXO-UL8 implements decoupled 1D virtual dynamics for each joint, hard and soft bounds are simultaneously shown on the elbow (joint 4) and shoulder rotation (joint 3) DoFs. The hard bound restricts the elbow motion to the range of $[0, 70°]$, whereas the soft bound on the shoulder joint becomes active once the motion exceeds $[15°, 60°]$. The parameters of the soft bound are $k_s = 10$ N/m, $d_s = 0$ Ns/m. A movement trajectory, as shown in Fig. 4.5, is designed to exercise both bound types in the joints.

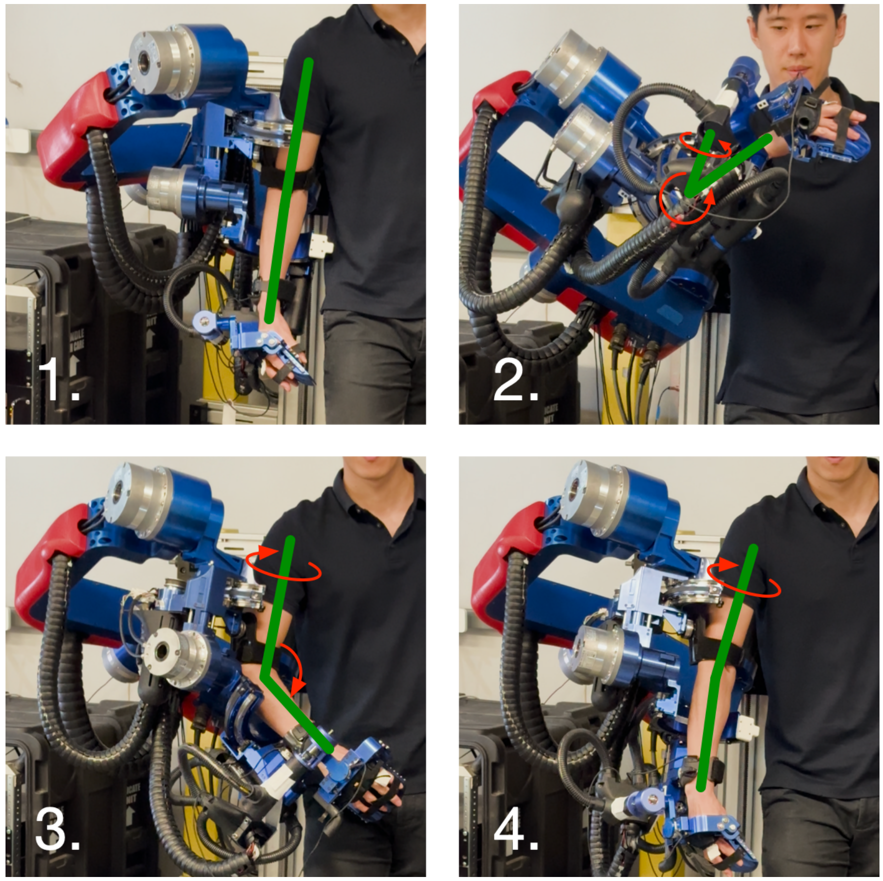


Figure 4.5: The bounds experiment on the EXO-UL8 has the operator follow a target trajectory consisting of 4 way-points (1-4). These way-points are strategically placed to engage shoulder rotation (joint 3) and elbow flexion (joint 4) near bound limits. Soft bounds restrict the shoulder position and hard bounds restrict the elbow position. The operator traverses through the way-points once, pausing at each for approximately 1 s. The corresponding motions from the previous way-point are indicated by red arrows, while the configuration of the human arm at each way-point is shown by a green line.

4.4.2.2 Collision Avoidance

While our collision avoidance method applies identically to dynamic avoidance of free entities and avoidance of static fixed entities, here we demonstrate only with a static entity on each of the two systems for clarity of results. With this methodology, the only difference between dynamic and static entity avoidance is the inputs of entity locations prior to collision avoidance algorithm being applied. For the V-Rex, the segment is physically represented by the right arm being static, which is oriented with the elbow-to-wrist link nearly vertical. This vertical segment is located at $(0.3 \text{ m}, 0.014 \text{ m})$ as shown in images (1) and (2) of Fig. 4.10. The operator initially aims to move the end effector of the left arm to pass in front of the static right arm. Next, the operator aims to move as far left from the static right arm as they can go before returning to the starting location. For ease of data representation, motions of the end effector and elbow-to-wrist link of the left arm are kept in a plane parallel with the ground, similar to the bounds experiment. The collision detection threshold is 0.27 m.

For the EXO-UL8, the axes of rotation for shoulder flexion and elbow flexion are first aligned. The other five joints are locked so that the motion of the arm is restricted to the plane perpendicular to these two unrestricted axes of rotation. The quadratic program of equation (4.11) includes these five locked joints as the constraints: $0 \leq \dot{\theta}_{r,i} \leq 0,\ i \in \{1, 3, 5, 6, 7\}$. A virtual collision segment is then placed in front of the arm at $(0.4 \text{ m}, -0.43 \text{ m})$ and oriented perpendicular to this plane, as shown in Fig. 4.6. The operator is instructed to move the arm above the collision segment and then return to the original location. The collision detection threshold is 0.1 m.

The collision avoidance algorithm requires the free entity positions to be updated on each iteration. For the V-Rex and EXO-UL8, free entities are the robotic manipulators. In each system, the relative locations of the manipulator bases are known, so the absolute locations of any point along the manipulators can be determined through forward kinematics. Thus, no calibration phase is necessary. For fixed entities, such as structural elements of the frame, their static locations are known (from design), so they can be added to the collision algorithm

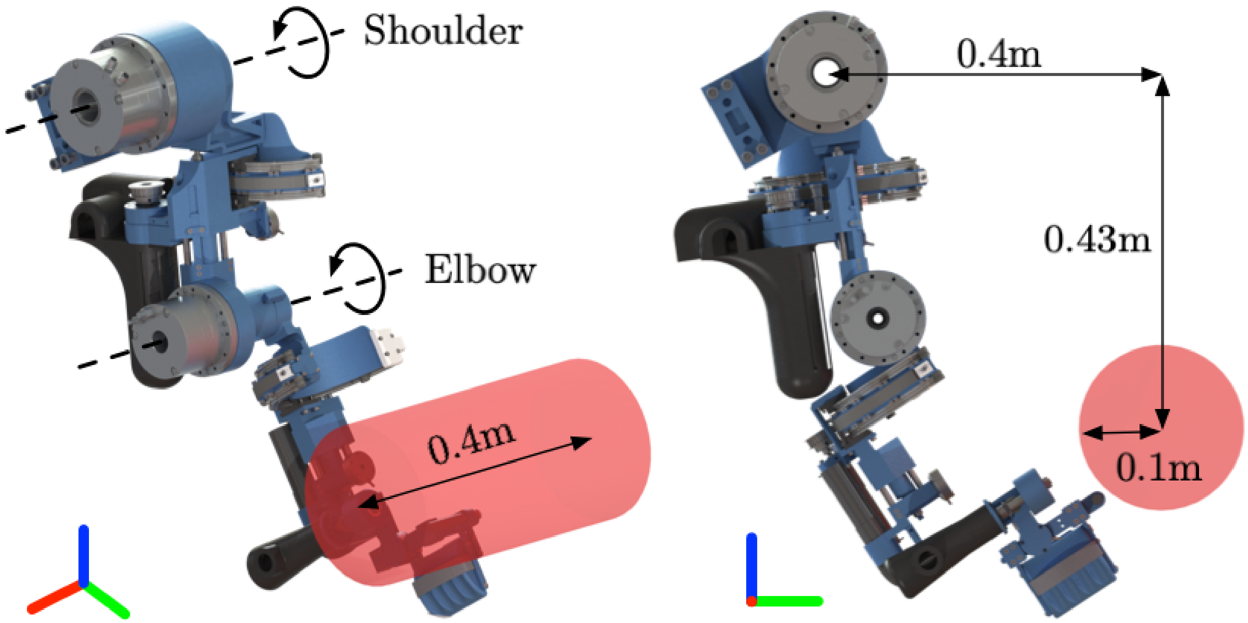


Figure 4.6: For the collision experiment on the EXO-UL8, the shoulder and elbow flexion joints are aligned and allowed to move, while all other joints are locked. A virtual fixed collision entity is placed in front of the arm and oriented to be parallel to the joints' axes of rotation. Collision avoidance distance is set to 0.1 m.

at initialization and do not need to be updated.

4.5 Results

4.5.1 Bounds

4.5.1.1 V-Rex

Fig. 4.7 presents the operator's trajectory as determined by the virtual dynamics with a hard bound. The trajectory took approximately 12 seconds to complete, resulting in an average speed of 10.2cm/s. The results demonstrate the effectiveness of a hard bound in limiting motion to the bounded region in task space, without destabilizing the system. Despite exerting forces in a direction to break outside the bounds, the virtual position is constrained and instead slides along the virtual bounding surface in compliance with the applied force. Conversely, when inside the bound, motion remains unrestricted as expected.

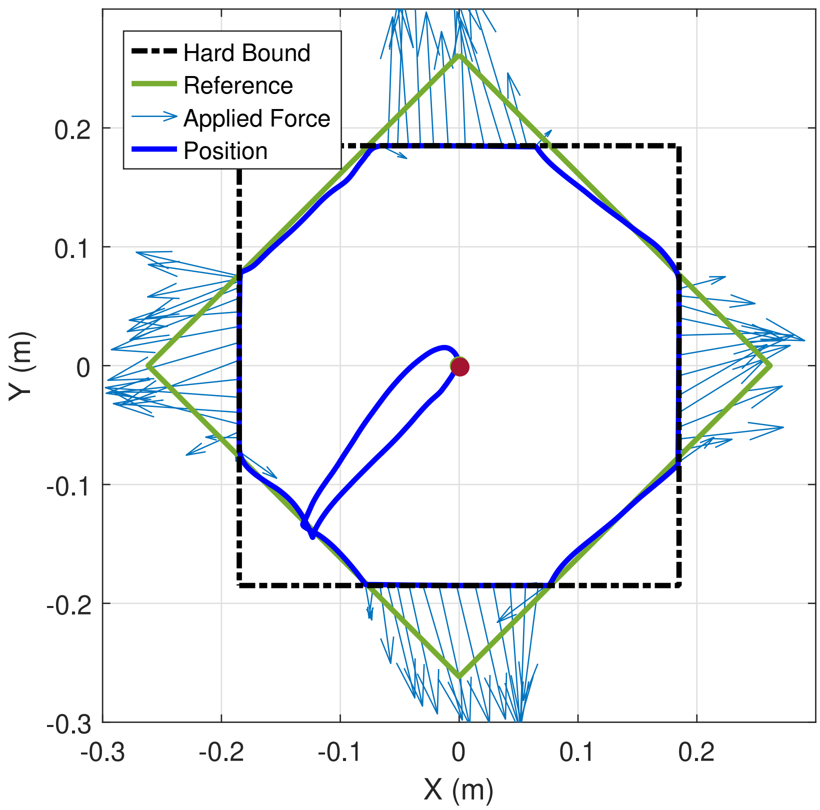
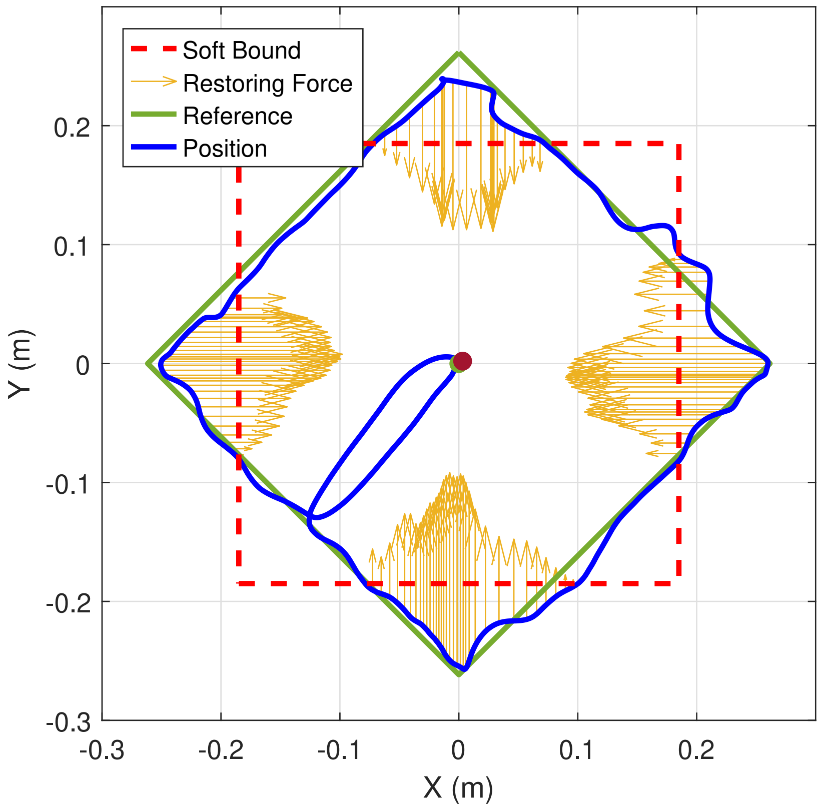


Figure 4.7: On the V-Rex, the operator's trajectory (blue line) tracking the reference (green) is impeded by the hard bound (dotted black). At the bound, the position is restricted and slides along the surface, despite the user's exerted force to move outside (blue arrows). Note that for clarity the forces are only shown when the position is at the bound. In contrast, while inside the bound, the virtual dynamics enable the operator to track the trajectory with high fidelity.

Fig. 4.8 illustrate the operator's trajectory and soft bound restoring forces. When inside the bound, no restoring forces are present, indicating unrestricted motion. However, upon breaking a bound, restoring forces are generated with a direction to move the operator back into bounds. Notably, all restoring forces are normal to the broken bound surface, and their magnitude increases with distance past the bound, consistent with the virtual spring-damper. The trajectory loop took approximately 18 seconds to complete, resulting in an average speed of 8.2cm/s.

4.5.1.2 EXO-UL8

The elbow flexion and shoulder rotation reference trajectories are plotted in Fig. 4.9, with the four way-points labeled. The trajectory was completed in approximately 20s, resulting in an average speed of $11°/s$. A soft bound, which is only present for the shoulder, produces restoring forces generated by a virtual spring whenever the trajectory exceeds the boundary. Note that the soft bound is only configured with a virtual spring, so the restoring force depends only on position. Hard-bounds were implemented in both DoFs, but only the elbow reached the limits due to the absence of an additional soft-bound. The applied forces at the hard-bounds show that the dynamics were only allowed to propagate tangentially to the bound surface, as designed. This experiment shows a typical application in which a soft bound is placed inside a hard bound. Motions towards bound limits are gently slowed while ensuring that the absolute limits are respected (e.g., at point 3 of Fig. 4.9).



H

Figure 4.8: On the V-Rex, the operator's trajectory (blue line) tries to track the reference (green). However, the presence of a soft bound (dotted red) results in a restoring force (yellow arrow) that amplifies with distance from the bound. As the operator navigates around the corners of the reference, the restoring force shifts from being fully antagonistic to providing some assistance in the direction of motion. This sudden change is reflected in the lower accuracy of the operator's trajectory when compared to the hard bound trials.

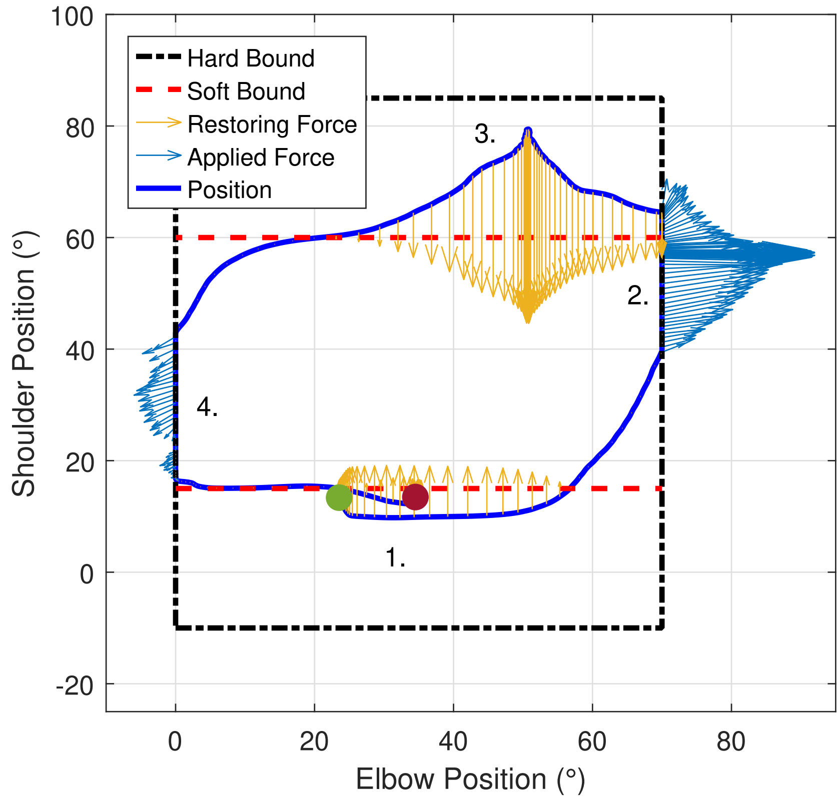


Figure 4.9: On the EXO-UL8, the operator's trajectory (blue line) starts at point 1 (green dot) outside the shoulder's soft bound (dotted red line), resulting in restoring forces (yellow arrow). While moving towards point 2, a hard bound (dotted black line) blocks the elbow, despite human-applied forces (blue arrow) pointing out of the bound, which are only plotted when the position is at a hard bound for clarity. The shoulder soft bound applies another restoring force towards point 3. Since the soft bound is configured as a virtual spring, the restoring force is only a function of displacement past the bound. At the intersection of the shoulder upper soft bound with the elbow upper hard bound (between points 2 and 3), the soft bound restoring force appears to suddenly be present. However, this is not the case; as the trajectory slides up along the elbow hard bound, soft bound restoring forces increase continuously. The largest force arrow visually occludes the smaller arrows underneath. At point 4, another hard bound restricts the elbow. The trajectory ends at the maroon dot back at point 1.

4.5.2 Collision Avoidance

4.5.2.1 V-Rex

The end effector's trajectory and the elbow-to-wrist link's pose during key collision avoidance points are shown in Fig. 4.10. At the first instance (1) of collision, the end effector trajectory is restricted such that the link will not continue into the collision detection region, despite the operator's force in that direction. Similarly at the second instance (2), the end effector trajectory is restricted; however, the restriction is now in a direction away from the collision detection region, since continued motion in the -X direction would result in the left elbow colliding with the right arm. Unlike the EXO-UL8, the operator's arms are not aligned with the manipulator, which could result in potential collisions between the two. However, the operator can be modeled as an uncontrollable free entity, so that the robotic manipulators would avoid collisions with it in a similar fashion.

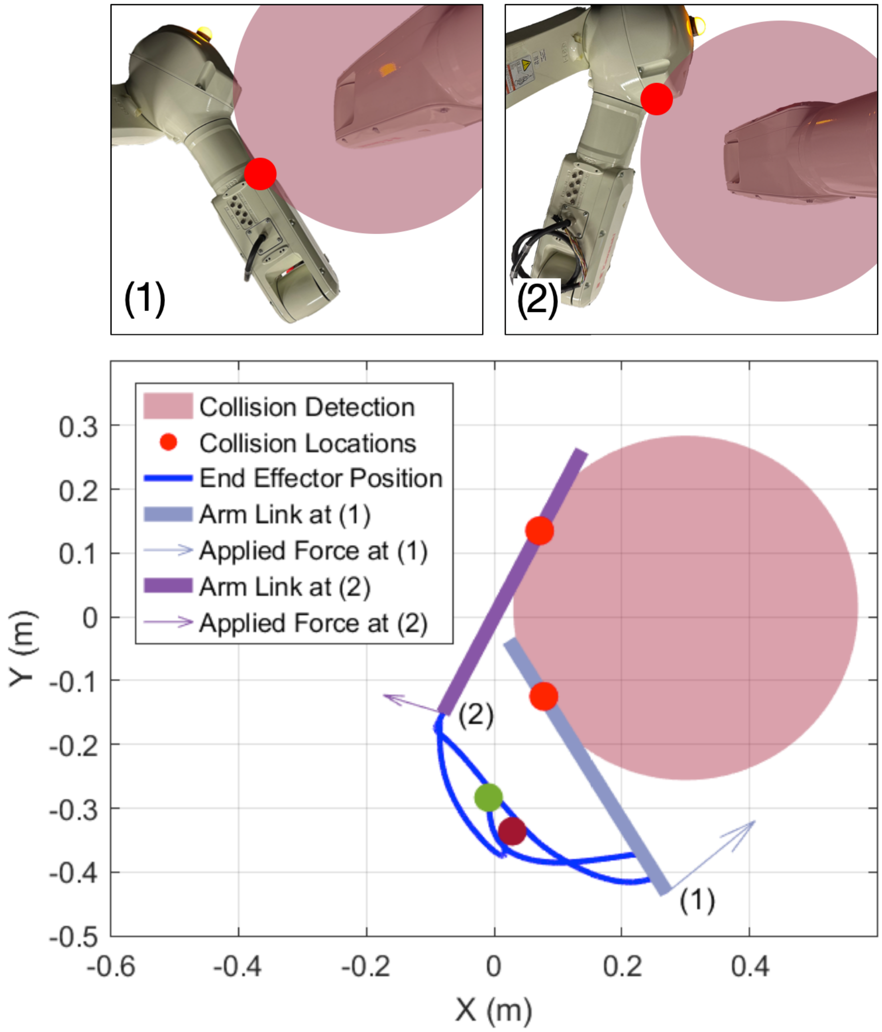


Figure 4.10: On the V-Rex, the operator's trajectory (blue line) starts at the green dot and ends at the maroon dot. During the trajectory, two collisions are avoided with a detection radius around the right arm (pink region). A physical representation of the data at the two time instances ((1) and (2)) is shown in the subfigures above. At instance (1), motion is restricted in the direction of the right arm, even though the operator is applying force to continue in that direction. At instance (2), motion is restricted from continuing in the -X direction despite applying force in that direction, because the elbow would collide if the motion were unrestricted.

4.5.2.2 EXO-UL8

The contour of the collision detection region of Fig. 4.6 is plotted in the shoulder and elbow flexion joint space in Fig. 4.11. The trajectory and human-applied forces are overlaid to show the motion being constrained from entering the collision region, even when the operator is pushing into it. Near $(63°, 9°)$, the trajectory doubles back slightly, but collision constraints are respected. Once the operator flexes their elbow, the region is avoided (from $(30°, 38°)$ upwards), and subsequent motion is unaffected. The experiment demonstrates collision avoidance's expected function.

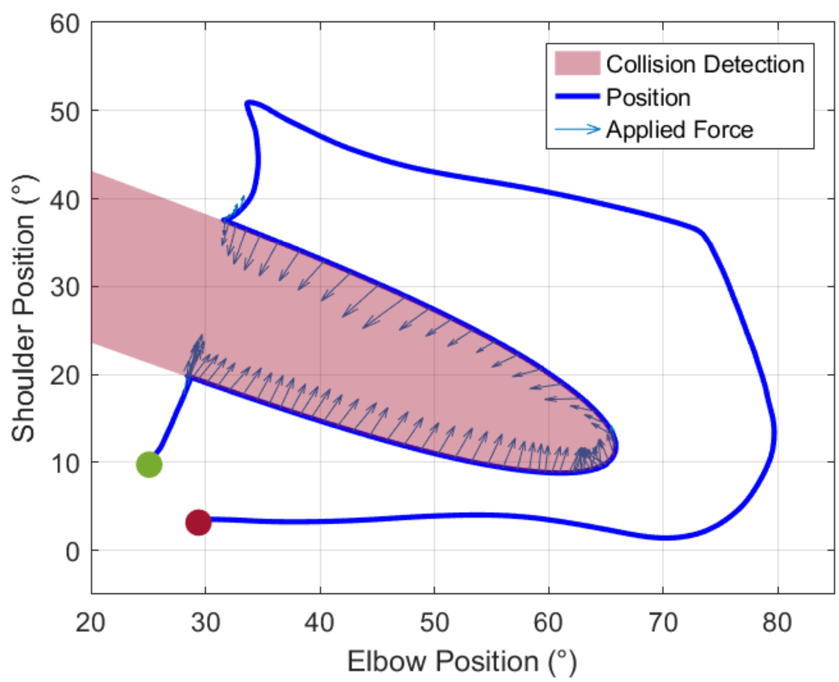


Figure 4.11: The virtual segment's collision detection region is transformed through inverse kinematics of the EXO-UL8 into the joint-space of the shoulder and elbow flexion DoFs. Even though human-applied forces try to push the trajectory into the region, it respects the collision constraints. Human-applied forces are only plotted at the boundary for clarity.

4.6 Discussion

4.6.1 Comparison to Existing Methods

In general, using runtime as a metric for comparison poses several issues. The implementation of the methodology has a significant influence; for instance, using a compiled language can be faster than an interpreted one, and the use of parallelism or vectorized operations can make the program even faster. As not all the studies in this comparison [66, 92, 94, 95, 97] provide code implementation or even specify the optimization solver used, a comparison based on runtime alone may not be feasible or meaningful. Furthermore, methods such as [97]'s utilize a neural net to solve the optimization. While this could be fast, it would require more effort to train and setup. Whether this trade-off is justified depends on the particular use case. The most notable aspects of collision avoidance that we have identified are: (1) modeling, which refers to the use of simpler shapes, known as primitives, used to represent the manipulators' geometry, (2) enumeration of primitives, which is the process of identifying potentially colliding primitives, (3) distance between primitives, which computes the shortest distance between two primitives as well as the locations on the primitives at which this occurs, (4) partial Jacobian, which relates the velocities of these locations to the manipulator's velocity, and (5) optimization solver, which is the numerical method or tool used to complete the computation. A comparison summary is presented in Table 4.1.

Collision avoidance is a complex problem that requires several application-dependent trade-offs to be made, which is why many algorithms exist. For instance, [66] uses meshes to represent the manipulator, which can be highly accuracy, but come at the expense of requiring a dedicated library for computing contact. On the other hand, studies like [94] and [95] represent the manipulator as a union of spheres. Although fast to compute pairwise distance, the method cannot find exact collision points, and requires a partial Jacobian to be precomputed for each sphere at setup, detracting from ease of use. The neural net of [97] makes an even bigger trade-off between speed, ease of use, and potentially even correctness.

Aspect\Study	Bosscher [92]	Liu [94]	Lin [95]	Zhang [97]	Todorov [66]	**Ours**
Modeling (location accuracy)	Spherical shells =	Spheres −	Spheres −	Line segments =	Meshes +	Line segments =
Enumeration of Primitives (complexity)	Pairwise with pruning =	Pairwise −	Pairwise −	Pairwise −	Pairwise with pruning =	Pairwise with pruning =
Distance between Primitives (complexity)	Unspecified Unknown	Analytic =	Analytic =	Iterative −	Iterative −	Analytic =
Partial Jacobian (setup)	Unspecified Unknown	Analytic for each sphere −	Analytic for each sphere −	Unspecified Unknown	N/A	Analytic for any point [98] =
Optimization Solver	Simulink	Unspecified	Simulink	Neural network	Newton	OSQP [103]

Table 4.1: The comparison examines the main aspects of collision avoidance using a suitable metric approach across five different recent publications. In each aspect, the relevant metric (shown in parenthesis) is assessed relative to our proposed approach and reported with one of the symbols: $\{+, -, =\}$, to indicate more performant, less performant, or comparable, respectively. The "unknown" keyword indicates that a comparison to our approach could not be made.

Our strategy prioritizes generality and ease of use in collision avoidance by using simple geometries and optimizing where possible (e.g., primitive enumeration). In addition to providing detailed specifications, such as the use of OSQP for speed and software compatibility (refer to [103] for its comparison results), our entire approach is implemented as a free and open-source library. This comparison aims to underscore the distinctions in our approach, allowing implementers to make an informed decision for their specific applications.

4.6.2 Potential Limitations

In situations where a robot manipulator operates at high speeds, it becomes impractical for the robot to come to an immediate stop when encountering a boundary, even if that boundary is defined as a hard bound. The proposed strategy focuses on creating trajectories that prioritize safety awareness, however the robot's overall ability to precisely track these trajectories is also determined by its controller's capabilities. Incorporating hardware or controller level information back into the reference generation can have potential improvements, such as in *model predictive control* (MPC). However, doing so creates an additional signal feedback loop from state to reference, which may affect overall stability. In general, the feedback interconnection of stable systems is not necessarily stable. So, a hardware-agnostic method that incorporates hardware level information while always guaranteeing safety may not be possible. To ensure generalizability of the approach, we chose not to allow for hardware-level information to influence the reference generation. Doing so would detract the approach's versatility, despite potentially achieving better results on specific systems.

A control rate of 1KHz is commonly recommended for reliable haptic interaction, but is not a strict rule and depends on specific factors such as desired virtual stiffness and the robot's mechanical capabilities. The haptic rendering presented only focused on soft bounds, which were not overly stiff. High-performance Kawasaki industrial manipulators of the V-Rex enabled convincing virtual force emulation at a lower control rate. Therefore, a control rate of 500Hz was chosen for the V-Rex as a suitable compromise between haptic rendering fidelity and CPU usage.

4.6.3 Choice of Parameters

The V-Rex virtual dynamics parameters were chosen empirically to simulate a 10 kg object in a damping medium of 15 Ns/m. This mass, though seemingly heavy, is not difficult to move in the absence of gravity. The damping constant was selected based on user feedback, as a compromise between transparency and dissipative stability. Soft bounds were configured with a spring constant of 250 N/m and damping of 60 Ns/m, in order to provide a noticeable resistance to the user.

In our implementation, we opted for rectangular and norm bounds for their intuitive nature. Rectangular bounds are easy to understand and configure, while norm bounds, though less common for position, prove useful for velocity by representing speed limits. However, the methodology accommodates any convex bound. In out experiments, position bounds were centered at the robot's starting point for clarity and to allow movement in any direction. In general, they can be placed anywhere containing the starting point.

4.6.4 Safety at Hard Bounds

When the virtual position reaches a hard bound, human-applied forces perpendicular to the bound do not impact virtual motion (see Fig. 4.7 and Fig. 4.9). However, these force components are still resisted by the robot's controller, potentially leading to increased motor currents. Safety implications are examined in detail for the shoulder interior/exterior joint of the EXO-UL8, which has the smallest gear ratio. Fig. 4.12 shows the motor current when the operator pushes against the hard bound at 55°. In this case, the current remains within the motor's continuous operating range (RE50 series 578298 from Maxon Motor). Although safety can be addressed at the reference generation level, hardware remains crucial for overall pHRI safety. The proposed methodology, being hardware agnostic, enhances safety at the reference generation level, without subjecting the robot's low level controller to any situation more dangerous than tracking a continuous position trajectory. However, controller-level safety must be ensured by the implementer, otherwise any algorithm would be subject to the same potentially dangerous situations.

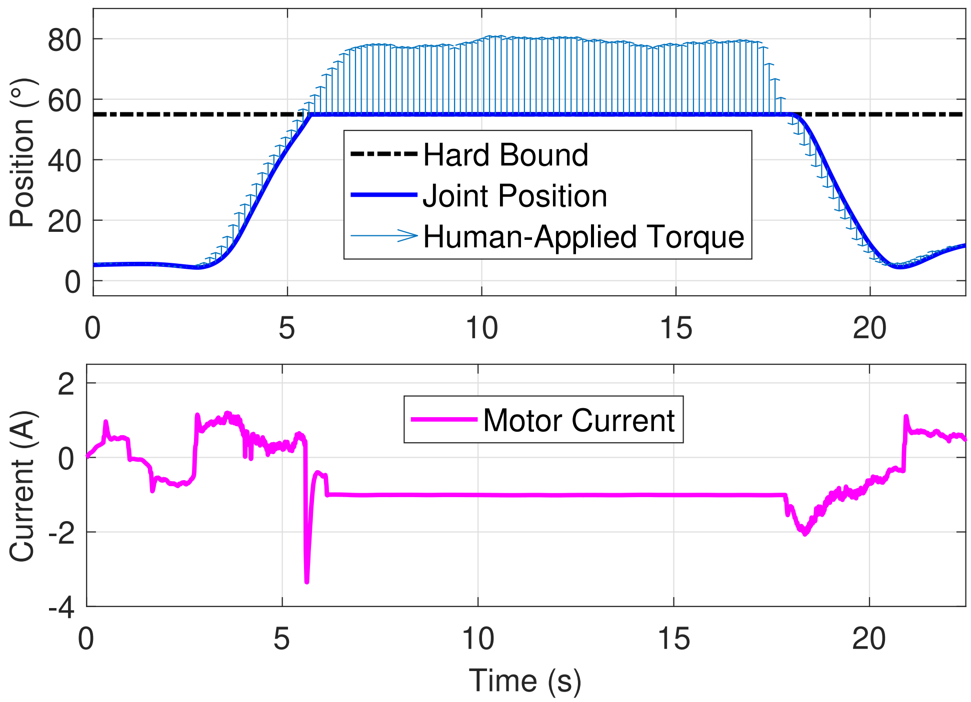


Figure 4.12: A hard bound at 55° on the shoulder interior/exterior rotation joint of the EXO-UL8 resists the operator's attempts to push past it. The commanded motor current (-1.01A) remains in its maximum continuous operation range of ±4A. Prior to reaching the position reaching the bound, the motor current varies due to the computed torque controller generating torques to compensate for reaction torques from motions of the other joints. However at the hard bound, the current appears flat as all joints are stationary. Friction at the joint and low back-drivability limit the variation in the human-applied torque from being easily seen in the current profile. In general, pHRI hardware should ensure that all currents always remain in safe operating ranges, despite adversarial attempts by the operator to cause unsafe situations.

4.7 Conclusion

This study presents a comprehensive safety-focused admittance control approach, consisting of soft virtual bounding regions, emulation of infinitely stiff bounds, and collision avoidance at any point along the manipulator. Experimental results validate the effectiveness of hard and soft bounds by confining trajectories to predefined regions, and producing restoring forces to return the reference to within the bound, respectively. They also validate the multi-arm collision avoidance methodology on both systems by successfully restricting trajectories that would cause any links of the serial manipulator to collide, despite the difference in control and redundancy.

Future directions on this framework plan to incorporate the human operator as a free entity in the collision avoidance algorithm. This would further improve safety by ensuring that the robot manipulators also cannot collide with the operator. The time-varying location of the operator's body can be determined by a vision-based system, which would likely require calibration to account for varying lighting conditions.

For utilization of our approach in existing and future systems within the pHRI community, an implementation as a free and open-source templated C++ library is available at: `https://github.com/jianwei-sun/gtfo`. The library requires a C++17 compiler and depends on Eigen [104] and OSQP [103]. Future work completed on this framework will be made available within the open-source library.

CHAPTER 5

Safety via Rate-Limiting

[4] **J. Sun**, P. W. Ferguson and J. Rosen, "Suppressing Delay-Induced Oscillations in Physical Human-Robot Interaction with an Upper-Limb Exoskeleton using Rate-Limiting," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, 2022, pp. 6695-6701, doi: 10.1109/IROS47612.2022.9981943.

5.1 Overview

In pHRI enabled by admittance control, delay-induced oscillations arising from both the neuromuscular time-delays of the human and electromechanical delays of the robot can cause unsafe instability in the system. This study presents and evaluates rate-limiting as a means to overcome such instability, and provides a new perspective on how rate-limiting can benefit pHRI. Specifically, a rate-limited and time-delayed HITL model is analyzed to show not only how the rate-limiter can transform an unstable equilibrium (due to time-delay) into a stable limit-cycle, but also how a desired upper-bound on the range of persistent oscillations can be achieved by appropriately setting the rate-limiter threshold. In addition, a study involving 10 subjects and the EXO-UL8 upper-limb exoskeleton, and consisting of 16 trials - 4 rate-limiter thresholds by 4 time-delays - is performed to: (1) validate the relationships between time-delays, rate-limits, and position bounds on persistent oscillations, and (2) demonstrate the effectiveness of rate-limiting for recovery from delay-induced oscillations without interfering with regular operation. Agreement of experimental results with the theoretical developments supports the feasibility of incorporating rate-limiting in admittance-controlled pHRI systems as a safety mechanism.

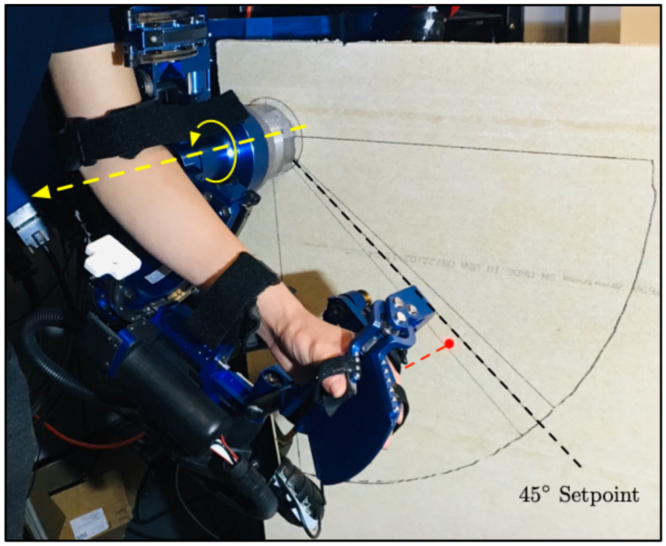
5.2 Introduction

pHRI is a necessary component of any exoskeleton-assisted physical therapy. Often enabled through admittance control, such pHRI allows the exoskeleton to fluidly follow the motions of the human operator. This functionality allows the exoskeleton to precisely apply assistive forces to the operator in order to create an effective training environment. Naturally, safety is a primary concern when interacting with powered robotic devices, as the target audience of such physical therapy often has limited physical capability.

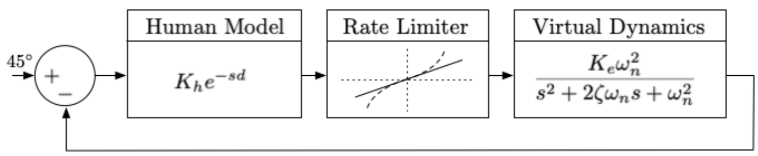
Even though the exoskeleton's admittance controller renders it a stable system by itself, the feedback connection of the human with the exoskeleton, along with neuromuscular delays of the human and electromechanical delays of the robot, can cause dangerous unstable oscillations. As discussed in [105], a human's natural tendency is to stiffen in order to suppress oscillations, but this can actually increase instability. An intuitive response to handling the instability is to add low-pass filtering, which assumes that removing high frequency components can bound velocity. However, not only is this false, low-pass filtering also introduces $-(\pi/2)n$ radians of phase lag for high-frequency components, where n is the filter's order. This phase lag reduces the system's phase margin and responsiveness to the operator, and can even be destabilizing [105]. Thus, ensuring stability in these HITL systems is non-intuitive, and must be addressed from the exoskeleton's point of view.

Various studies have proposed different methods of ensuring safety in pHRI, including, but not limited to: ensuring passivity of the system [78,106–109], reducing phase-lag through feedforward control [28, 110, 111], saturating forces/torques [82, 83], using adaptive control to ensure robustness against modeling uncertainties [112], and dynamically adjusting admittance control parameters [67, 68, 81, 84, 113].

Motivated by nonlinear control theory, passivity-based controllers ensure stability by preventing the feedback system from accumulating unbounded energy. Previous literature shows the utility of passivity-based approaches, such as in designing stabilizing controllers for exoskeletons [106,108], mitigating communication delays in teleoperation [107], and over-



(a) Experiment Setup



(b) Human-in-the-Loop Model

Figure 5.1: (a) Experiment setup using the left elbow of the EXO-UL8 upper-limb exoskeleton. Subjects must track the 45° setpoint for various values of time-delay and rate-limiter thresholds. A laser pointer (enhanced in red) provides visual feedback to the subjects. (b) A block diagram of the HITL used in the theoretical developments of sections 5.3 and 5.4.

coming modeling uncertainties [109]. While passivity-based control is effective for ensuring stability, the controller requires knowledge of the passivity properties of the human, which may be difficult to precisely determine.

Another technique for improving stability is to reduce phase-lag in the HITL system by utilizing disturbance observers to provide feedforward compensation [28, 110, 111]. For these systems, the controller requires measuring or estimating the robot's acceleration, which can be susceptible to modelling errors. A simpler method is to just saturate the human-applied forces by limiting the maximum values measured. While this can prevent divergence in some cases, it is inadequate in general because: (1) it is unclear how the saturation threshold

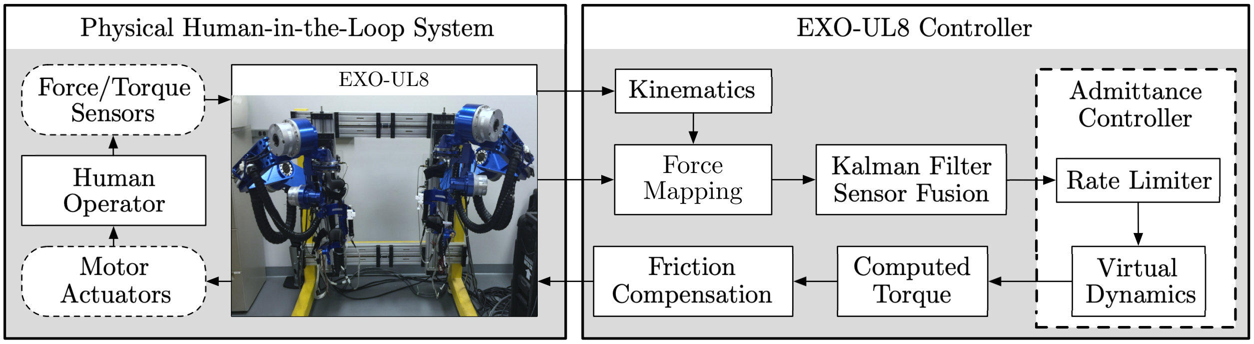


Figure 5.2: Block diagram of the EXO-UL8's control architecture with the admittance controller shown in the dash-outlined box on the right. The rate-limiter outputs the filtered estimated human-applied torques, which propagate the virtual dynamics. The trajectories of the virtual dynamics are then tracked by the computed torque controller, which also compensates for gravity and link inertia. On the physical side, the operator interacts with the EXO-UL8 through force/torque sensors and motor actuators, which are shown in the dashed round blocks but are considered as part of the EXO-UL8.

should be set to prevent instability without interfering with regular high-speed interaction, (2) saturation does not prevent high frequency switching between threshold values, and (3) it may result in the "wall-sticking" problem described in [78] which can produce undesirable performance.

Safety has also been addressed by detecting instability using various heuristics and then dynamically adjusting the admittance control parameters [67,68,81], but this requires tuning based on experimental data and may not generalize to other HITL robotic systems.

In this paper, we present a new perspective on rate-limiting as a safety mechanism against delay-induced instability. Rate-limiting is traditionally regarded as an undesirable nonlinearity since it is difficult to compensate for, as seen in early fighter jets [114, 115]. Subsequent studies have therefore investigated various other methods for prediction and compensation [116–118]. However, rate-limiting has useful properties that can benefit pHRI. In this work, we propose and experimentally validate the incorporation of a rate-limiting filter to our admittance controller to prevent delay-induced instability for the EXO-UL8 upper-limb exoskeleton. Even though the filter does not guarantee equilibrium stability, it prevents trajectories from diverging and can allow the human operator to recover from unstable oscillations. We believe that the rate-limiting filter can improve safety in admittance control-based pHRI because it does not require precise dynamical/passivity models of the human, does not require estimating acceleration, can overcome issues with high frequency switching apparent in force/torque saturation methods, avoids the need for any detection of instability, and is simple to implement. Our contributions are therefore:

1. Theoretically analyzing the rate-limiter in a HITL system, as shown in Fig. 5.1, and demonstrating how persistent oscillations due to large time-delays can be bounded as desired by selection of rate-limiter threshold,

2. Validating the rate-limiter on an upper-limb exoskeleton for 10 subjects to empirically demonstrate utility,

3. Demonstrating the effect of time-delay for the HITL system and how rate-limiting can prevent instability to allow for recovery from persistent oscillations.

5.3 Model of Human-in-the-Loop

The EXO-UL8 is a bimanual upper-limb exoskeleton comprised of a pair of serial manipulator arms, each with 7 active DoFs, developed to support research in pHRI and stroke rehabilitation. Admittance control is utilized to enable physical interaction by sensing human input using force/torque sensors located along the exoskeleton's arm and then converting it into equivalent torques at the exoskeleton's joints. These torques then drive virtual dynamic models consisting of decoupled second-order systems with low virtual masses, whose dynamics are then tracked by the exoskeleton's computed-torque controller, as shown in Fig. 5.2. The overall effect is that to the human, the exoskeleton appears to move according to the virtual dynamics. More information about the sensing and control architecture of the EXO-UL8 can be found in [5,6].

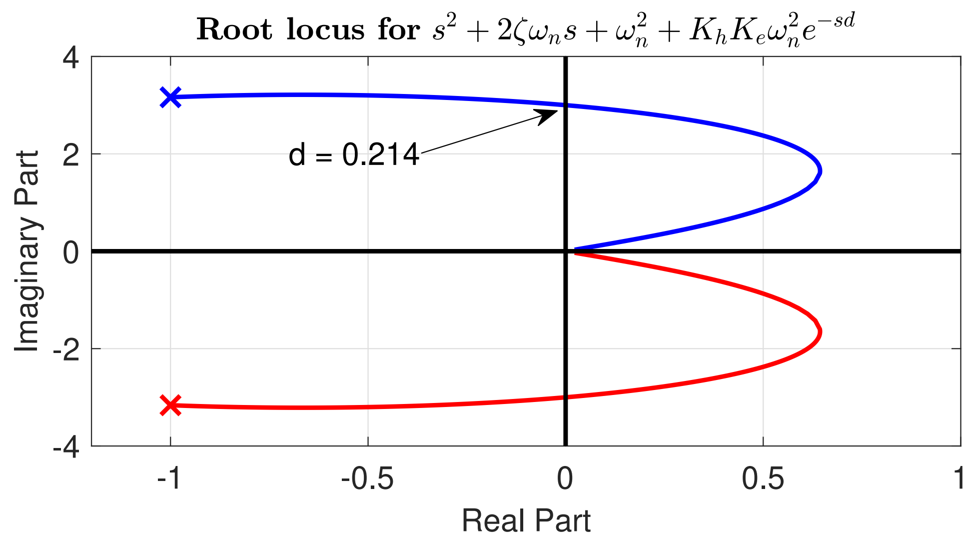


Figure 5.3: A root locus for the closed-loop system shows that the closed-loop poles cross the imaginary axis for a sufficiently large time-delay.

Consider the simplified HITL model consisting of a single second-order *linear time-invariant* (LTI) system (to represent a single DoF of the exoskeleton's virtual dynamics) in feedback with a static gain, as shown in Fig. 5.1. By design, the exoskeleton system is stable as its poles are located in the *open left half-plane* (OLHP) at $\{-\sigma \pm j\omega_d\}$, where $\sigma = \zeta\omega_n$, $\omega_d = \omega_n\sqrt{1-\zeta^2}$, ω_n and ω_d are the natural and damped frequencies of the virtual second-order dynamics, respectively, and ζ is the damping ratio. The human interaction is modeled as a static gain, similar to [115], with delay. The static gain represents human-applied forces trying to drive the exoskeleton to a constant setpoint (taken as zero without loss of generality by using error coordinates), while the delay models neuromuscular delay [41].

5.3.1 Instability due to Time-Delays

Even though both the human and the exoskeleton are stable systems individually, their feedback connection is not necessarily stable. This is shown by considering the stability of the closed-loop system as a function of the time-delay, $d \in \mathbb{R}_{\geq 0}$. As the closed-loop system is still LTI, stability can be assessed by its characteristic polynominal:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 + K_h K_e \omega_n^2 e^{-sd}. \tag{5.1}$$

A root-locus of the zeros of the characteristic polynominal is shown in Fig. 5.3 with sample values: $K_h = 10$, $K_e = 1$, $\zeta = 1$, $\omega_n = 1$. The branches cross the imaginary axis, indicating that a sufficiently large time-delay can destabilize the system.

5.4 Rate-Limiting Filter

Consider the rate-limiting filter added after the human input in Fig. 5.1. Also known as slew rate-limiting, the nonlinear filter places bounds on the maximum and minimum rates of change of the input signal. In continuous time, the rate-limiter can be implemented as a first-order filter with saturation:

$$\dot{y}(t) = \mathrm{sat}_R(pu(t) - py(t)), \tag{5.2}$$

where $R > 0$ is the rate-limiter threshold, $u(t)$ is the input scalar signal, $y(t)$ is the output scalar signal, and $p > 0$ is a constant chosen to be much larger than the other system poles. The saturation function is defined as:

$$\mathrm{sat}_R(x) := \begin{cases} -R, & \text{if } x \leq -R, \\ x, & \text{if } -R < x < R, \\ R, & \text{if } R \leq x. \end{cases} \tag{5.3}$$

5.4.1 Human-in-the-Loop System with Rate-Limiting

The rate-limiter is added to the HITL system as shown in Fig. 5.1 to limit the maximum rate-of-change of the human-applied torque. The rate-limiter's output depends on the input signal's frequency and amplitude, and also introduces additional frequency components not present in the input signal. To analyze the effect of the rate-limiter on the HITL system, it is useful to first determine the phase lag of the filter, which helps in calculating the period of the limit cycle in the closed-loop system.

5.4.1.1 Phase Lag of Rate-Limiter

Consider a sinusoidal signal of amplitude A and angular frequency ω input to the rate-limiter with rate-limiter threshold R such that:

$$A \geq \frac{\pi R}{2\omega}, \tag{5.4}$$

which ensures that the rate-limit is always active, as shown in Fig. 5.4. The output signal is a periodic triangular wave whose fundamental has the same frequency as the input, but with reduced amplitude and additional phase lag. The phase lag, ϕ, of the output's fundamental harmonic can be determined by considering the start time as the crest of the input and noting that the maximum value of the output is $\frac{\pi R}{2\omega}$. Then, the phase can be calculated by solving for ϕ in:

$$A\cos(\phi) = \frac{\pi R}{2\omega}, \tag{5.5}$$

to yield:

$$\phi = \cos^{-1}\left(\frac{\pi R}{2A\omega}\right), \tag{5.6}$$

which is a function of both input frequency and amplitude. Also unlike that of high-order linear filters, the phase lag of the rate-limiter is upper-bounded by $\pi/2$, occurring when inputs have high frequency and high amplitude. This difference demonstrates one intuitive reason for how a rate-limiter can prevent instability in the closed-loop system.

5.4.1.2 Limit Cycle Frequency and Bound on Position

The objective now is to compute the resonant frequency, ω, of the HITL system. To do this, first consider the LTI virtual dynamics of the exoskeleton system. Its phase lag is a function of the input signal's frequency, so the response of the system to a triangular wave input oscillating at ω can be computed. For an even periodic triangular waveform with frequency ω (and consequently, period $T = \frac{2\pi}{\omega}$) defined as:

$$x_T(t) := \begin{cases} \frac{RT}{4} - Rt, & \text{if } t \in \left[0, \frac{T}{2}\right), \\ -\frac{3RT}{4} + Rt, & \text{if } t \in \left[\frac{T}{2}, T\right), \end{cases} \tag{5.7}$$

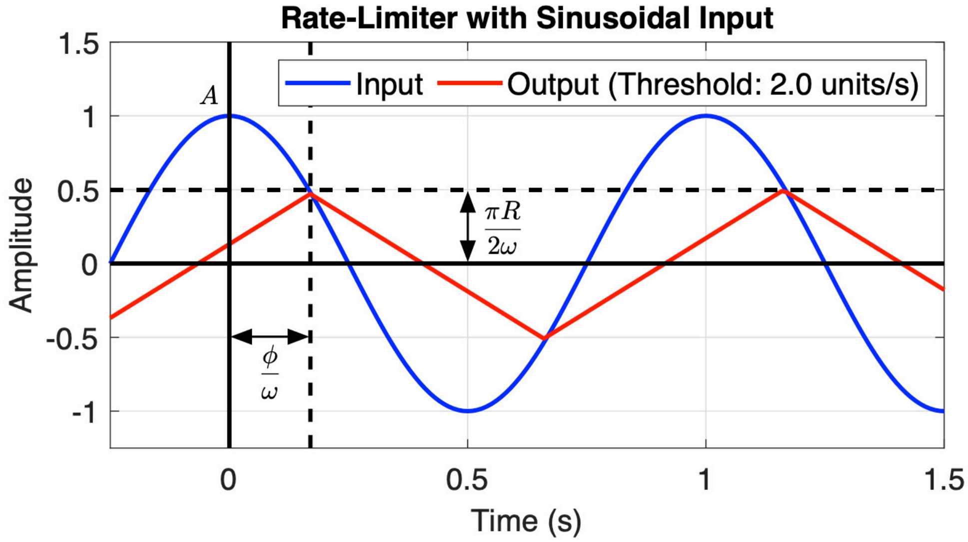


Figure 5.4: A sinusoidal input with amplitude greater than or equal to $\frac{\pi R}{2\omega}$ results in a triangular waveform output. In this example, the rate-limiter threshold is set to $R = 2$, and the input has amplitude $A = 1$ and frequency $\omega = 2\pi$. By inspection, the fundamental harmonic of the output is phase-lagged and has decreased amplitude. Furthermore, the triangular waveform has frequency components at multiples of the fundamental harmonics not present in the input signal, which is characteristic of the filter's nonlinearity.

it can be expressed as a sum of its harmonics:

$$x_T(t) = \sum_{n=-\infty}^{\infty} \left(\frac{\pi R}{2\omega}\right) c_n e^{jn\omega t}, \tag{5.8}$$

$$c_n = \frac{4\sin\left(\frac{\pi n}{2}\right)^2}{\pi^2 n^2}. \tag{5.9}$$

Let the Fourier transform of the second-order exoskeleton system be denoted as $G(\cdot) : \mathbb{C} \to \mathbb{C}$. Since it is LTI, its output for the input signal x_T is a superposition of the input's harmonics:

$$y_T(t) = \sum_{n=-\infty}^{\infty} \left(\frac{\pi R}{2\omega}\right) c_n \left|G(j(nw))\right| e^{j[n\omega t + \angle G(jnw)]}. \tag{5.10}$$

Since the phase lag of the rate-limiter also depends on its input signal's amplitude, it is helpful to compute the ratio of the input and output signal maximum amplitudes for G. The

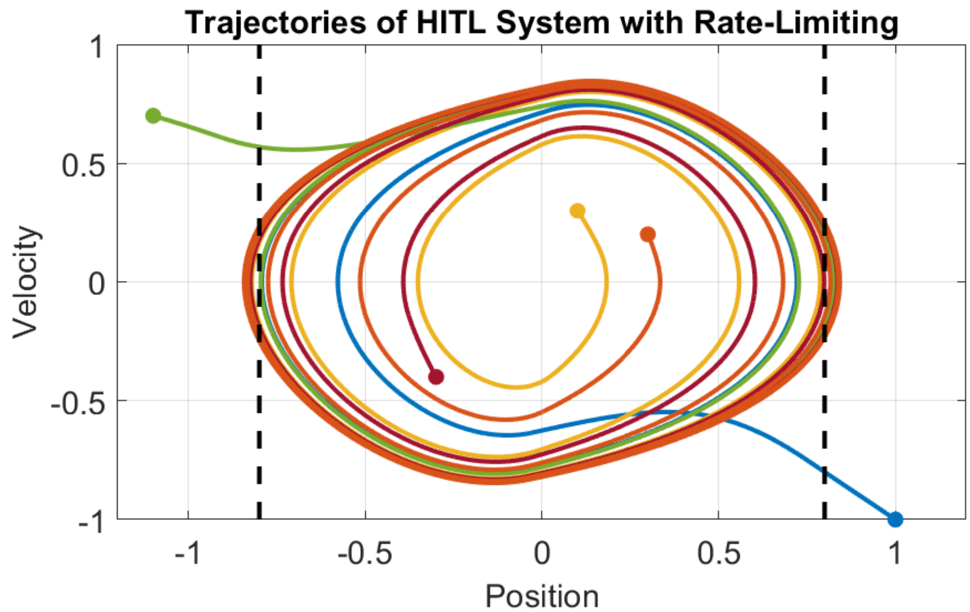


Figure 5.5: Trajectories of the HITL with rate-limiter threshold set according to equation (5.14) for $\epsilon = 0.8$ (shown as dashed lines). Since the closed-loop system is a delayed differential equation, trajectories can cross each other because the system's instantaneous state does not uniquely determine its derivative. Certain trajectories appear to exceed the desired ϵ boundary due to the decaying contributions of the initial conditions; ϵ is a bound on the steady-state oscillations. For all trajectories starting close to the limit cycle, their steady-state behavior approaches the stable limit cycle.

amplitude of the input triangular wave is $\frac{\pi R}{2\omega}$, but the output signal attains its maximum value at the delay corresponding to the phase lag of the fundamental harmonic, which is $-\angle G(j\omega)$. Hence, the ratio of input and output maximum amplitudes as a function of frequency, denoted by the function $r(\cdot) : \mathbb{R} \to \mathbb{R}$, can be calculated as:

$$r(\omega) := \sum_{n=-\infty}^{\infty} c_n \left| G(j(nw)) \right| e^{j[\angle G(jnw)) - \angle G(j\omega)]}, \tag{5.11}$$

which does not depend on the rate-limiter threshold, R. Next, to compute the resonant frequency, ω, of the HITL closed-loop system as shown in Fig. 5.1, consider the gain around

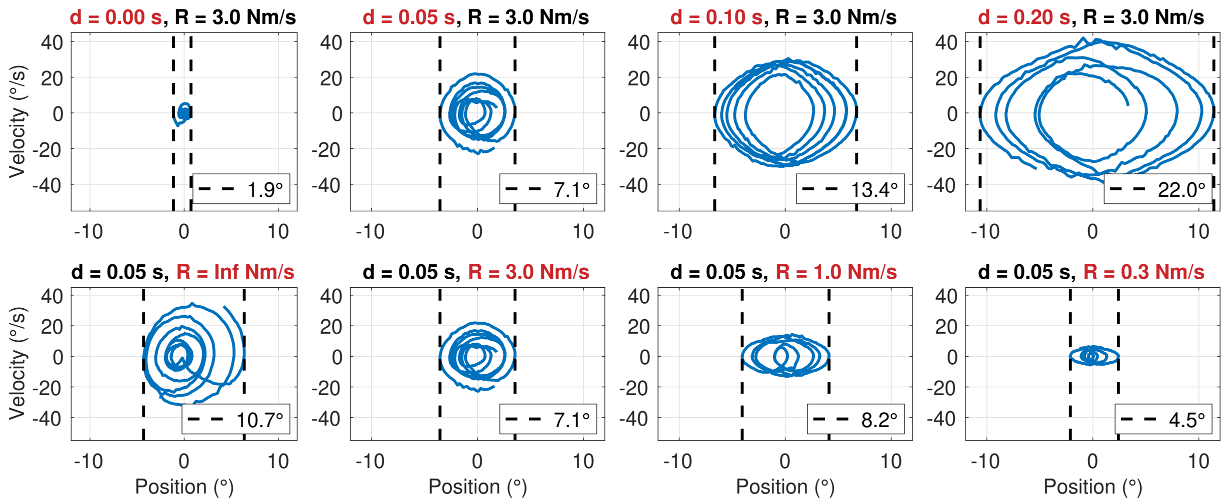


Figure 5.6: Zero-mean position and velocity trajectories of subject 1. The top row shows trajectories of persistent oscillations with increasing position range (range shown in the legends) as the time-delay value is increased (highlighted in red) for a constant rate-limiter threshold. The bounds increase monotonically with time-delay. The bottom row shows the effect of decreasing rate-limiter threshold for a fixed time-delay. In this case, the delay is mild and does not result in instability when the rate-limiter is not present (leftmost subfigure). As the threshold decreases, so does the position bound. However, the decrease is not always monotonic, as shown by $R = 1.0$ Nm/s. Simplifying assumptions, such as the human being modeled as a static gain with delay, may be inadequate in capturing complex time-varying behaviors that affect how the human responds to low rate-limiter thresholds.

the loop. If the rate-limiter's output amplitude is $\frac{\pi R}{2\omega}$, then this amplitude becomes

$$A = K_h r(\omega) \frac{\pi R}{2\omega}, \tag{5.12}$$

as the signal makes its way around the loop to the rate-limiter's input. When $K_h r(\omega) \geq 1$, the condition in equation (5.4) is satisfied, and the rate-limiter is always active. For persistent oscillations to occur, the phase lags contributed by the rate-limiter, time-delay, and the exoskeleton system also must result in one period of oscillation. This is described by:

$$\angle G(j\omega) - \cos^{-1}\left(\frac{1}{K_h r(\omega)}\right) - \omega d - \pi = -2\pi, \tag{5.13}$$

where the first term is the phase lag of the exoskeleton system, the second is from the phase lag of the rate-limiter by substituting equation (5.12) into equation (5.6), the third is from the time-delay, and $-\pi$ is from the negative input to the summation block. As an example, solving equation (5.13) numerically with the same parameters as in Fig. 5.3 and $d = 0.3$ s yields an oscillation frequency of $\omega = 0.955$ rads/s.

Once ω is determined, the output position can then be bounded to $\pm\epsilon$ by selecting a rate-limiter threshold of:

$$R \leq \frac{2\omega\epsilon}{\pi r(\omega)}. \tag{5.14}$$

Thus, the size of the stable oscillation can be set as desired for the given virtual dynamics of the system. Fig. 5.5 shows the stable limit cycle with the desired bounds. This analysis shows the feasibility of rate-limiting as a safety mechanism for human-exoskeleton interaction as unstable trajectories no longer diverge due to time-delay instability, but rather flow in a limit cycle, allowing the human to recover.

5.5 Implementation

The EXO-UL8's admittance controller consists of decoupled virtual dynamics at each revolute joint, and converts human-applied forces (inputs) into reference trajectories (output). Rate-limiters are placed between the human-applied forces and the virtual dynamics as shown in Fig. 5.2. A discrete-time implementation of the rate-limiter dynamics from equation (5.2) is given in equation (5.15), which uses the saturation function in equation (5.3). In the equation, u_k and y_k are the filter's input and output, respectively; Δt is the period of the software loop, and R is the rate-limiter threshold:

$$y_k = y_{k-1} + \text{sat}_{(\Delta t)R}(u_k - y_{k-1}). \tag{5.15}$$

5.6 Experiments

5.6.1 Experiment Setup

A series of setpoint tracking experiments are performed to validate the expected relationships between time-delay, rate-limiter threshold, and the bound on position for persistent oscillations. The experiment consists of 16 trials: 4 values of rate-limiter threshold for each of 4 values of artificially added time-delay. The experiments are performed on the left elbow flexion DoF of the EXO-UL8, as shown in Fig. 5.1. In each trial, the subject tries to maintain a constant elbow angle after the experimenter programmatically sends a fixed-magnitude impulse disturbance with random direction. The trial is concluded and marked as unstable if the interaction diverges due to instability. Otherwise, the subject holds the reference position for 15 seconds, as paced by a metronome. An overview of the experimental procedure is presented in Procedure 2. A total of 10 subjects (ages: 27.5 ± 2.6; 2 female, 8 male) participated in the study. The experiment is performed in accordance with IRB #18-00766.

5.6.2 Results and Discussion

The experiment aims to validate the rate-limiter on the EXO-UL8, and demonstrate that position ranges of persistent oscillations increase with higher time-delays, but decrease with lower rate-limiter thresholds. Experimental trajectories for a sample subject are plotted in Fig. 5.6. In this figure, increasing time-delay for a fixed rate-limiter threshold results in larger oscillations in steady-state. In contrast, decreasing rate-limiter threshold for a fixed time-delay yields smaller oscillations, suggesting that a lower threshold is needed if the HITL system has significant delay. For this subject, the position bound is slightly larger for $R = 1.0$ Nm/s than for $R = 3.0$ Nm/s; however, this discrepancy can be explained by inter-trial variability of the subject and limitations of the constant gain human model used in the theoretical development. To better evaluate the overall trends, consider the results averaged across all subjects in Fig. 5.7.

Procedure 2 Experimental Procedure

for each subject $s \in \{1, \ldots, 10\}$ **do**
    Subject familiarizes with EXO-UL8 for 5 minutes
    for each delay $d \in \{0, 0.05, 0.1, 0.2\}$ **do**
        for each threshold $R \in \{\infty, 3, 1, 0.3\}$ **do**
            Set d and R on EXO-UL8
            Subject moves to 45° position
            Experimenter sends $\pm$ impulse disturbance
            if tracking becomes unstable **then**
                Conclude trial
            else
                Subject moves to 45° position
                Subject holds for 15 seconds
    Subject reports on experience

Subject averaged results show that in order to mitigate the enlarging effects of time-delay on steady-state oscillations, the rate-limiter threshold should be reduced. It should also be noted that instability only occurred in trials in which the rate-limiter was inactive ($R = \infty$) and the delay was large ($d \geq 0.1$ s). This suggests that rate-limiting, even with a large threshold, can prevent instability, which was theoretically predicted by the existence of stable limit-cycles in the simulated trajectories of Fig. 5.5. Furthermore, the undelayed ($d = 0$ s) trials corresponding to $R = 3.0$ Nm/s and $R = \infty$ showed similar position ranges, indicating that the rate-limiter did not significantly affect normal operation. This is also supported by the percentage of time that the rate-limiter was active, as shown in Fig. 5.8. As delays become larger, so do the active times of the rate-limiter in order to overcome the increased likelihood of instability. However, during the undelayed trials, the rate-limiter is still partially active. This is likely a consequence of numerically differentiating the noisy human-applied torque signals, whose derivatives frequently contain impulses exceeding the rate-limiter threshold. By comparing the three different rate-limiter thresholds, it is apparent

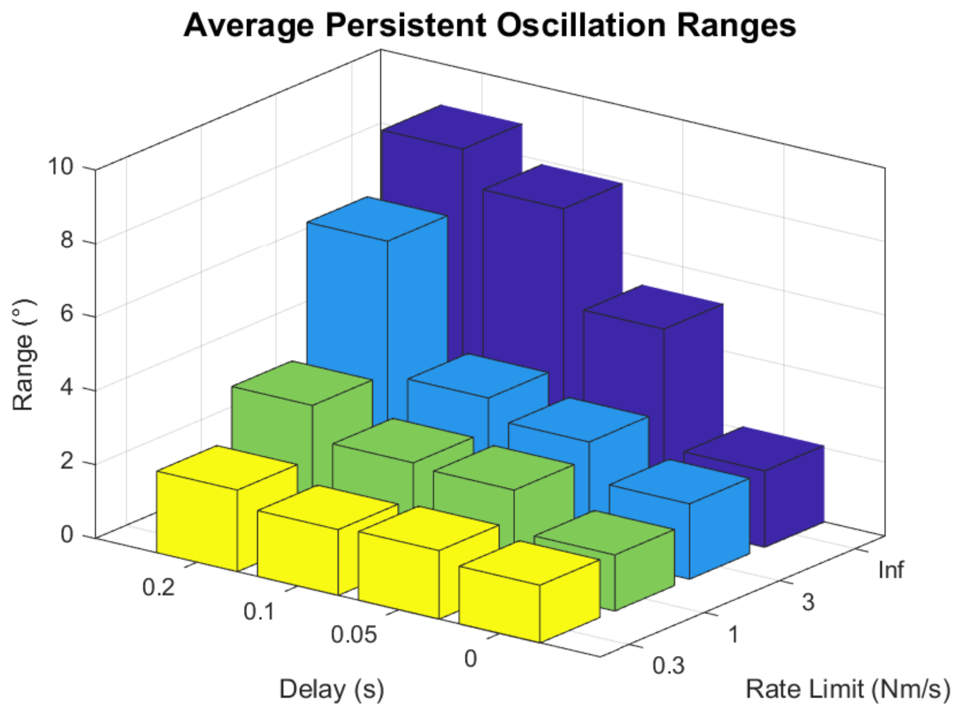


Figure 5.7: Position ranges corresponding to 5.5 s of steady-state oscillations of all trials averaged across all subjects. Trials in which subjects experienced instability are recorded as the range exceeding 30°; these only occurred in the absence of the rate-limiter and with delay ≥ 0.1 s.

that while a lower threshold value results in lower amplitude oscillations, as was seen in Fig. 5.7, too low a value may hinder regular operation.

5.6.2.1 Minimum Rate-Limiter Threshold

For a very small rate-limiter threshold ($R = 0.3$ Nm/s), the position range did not strictly increase with time-delay. Although equation (5.14) suggests that an arbitrarily small bound can be achieved by a sufficiently small threshold, the experimental results suggest a minimum threshold beyond which the relationship is no longer valid. The theoretical result assumed that the human can be modeled as a static gain controller with constant delay. While this assumption is sufficient in some cases, these experimental results show the extent to which that assumption is valid. For such a small rate-limiter threshold, the phase delay of equation (5.13) approaches its maximum, and makes the exoskeleton's motions feel sluggish, as described by subjects in the post-experiment discussions. This is also indicated by the rate-limiter being active 65% of the time at this threshold, even with no added time-delay, as shown in Fig. 5.8. The increased activity of the rate-limiter and its phase-lag affect the subjects' ability to track the setpoint and cause them to feel less in-control of the exoskeleton, which is a phenomenon not reported for trials corresponding to the larger rate-limiter thresholds. The qualitative and subjective results of the experiments suggest that although rate-limiting is beneficial for mitigating instability, there is a minimum threshold beyond which its limitations outweigh its benefits.

5.6.2.2 Generalization of Rate-Limiter Thresholds

The relationship between the rate-limiter threshold and the oscillation range depends on the virtual dynamics and associated delays of the admittance control system, which do not change across trials and operators; and the electromechanical delays of the operator, which are comparable across individuals [119]. Thus, specific threshold values need only be tuned for each pHRI devices by using a reasonable upper-bound for the operator's delay for the target motion.

5.6.2.3 Recovering from Instability

In a regular HITL system, the rate-limiter is not expected to always be active. Instead, its role is to act as a safeguard by preventing delay-induced oscillations from diverging, while minimizing interference with normal operation. An example of this behavior can be seen in Fig. 5.9. In this trial, the time-delay is large ($d = 0.2$ s) and destabilization occurs at the 14 s mark, but the rate-limiter prevents the oscillations from diverging. At 20 s, the subject stabilizes and is able to continue normal operation afterward. Causes of instability can include any combination of: the subject becoming fatigued; the subject

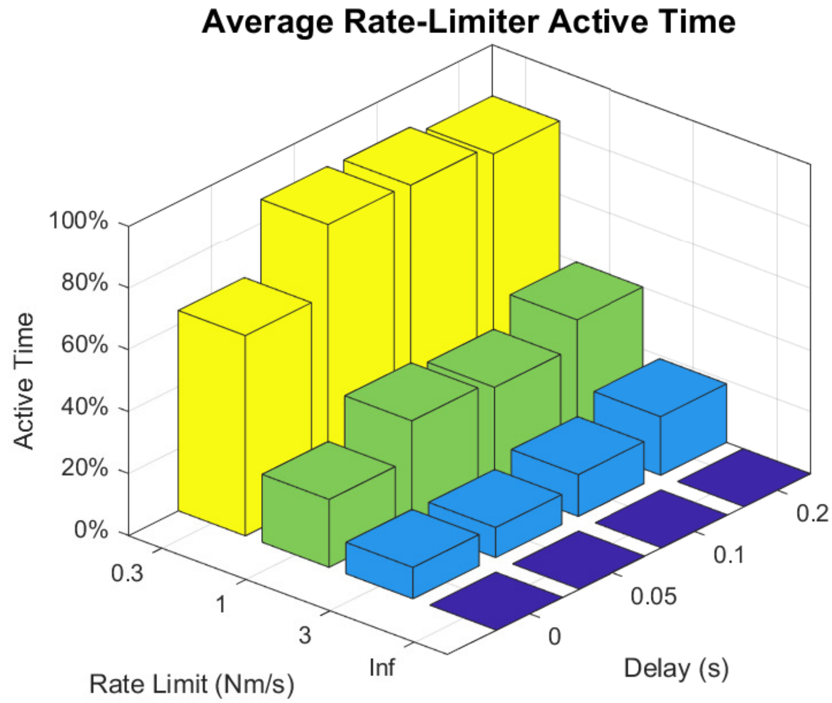


Figure 5.8: Active time of the rate-limiter for all trials averaged across all subjects. Greater time delays trigger the rate-limiter more frequently in order to prevent instability. However, longer active times in undelayed trials indicate that too low a threshold may inhibit normal operation.

tensing their arm in an attempt to fight instability [105]; or the subject losing concentration and relying purely on haptic feedback, which can be much less effective than utilizing both visual and haptic feedback [120, 121]. Such scenarios are not uncommon in HITL systems, which underscores the importance of safety from instability. These experimental results demonstrate the effectiveness of the rate-limiter as a safety mechanism.

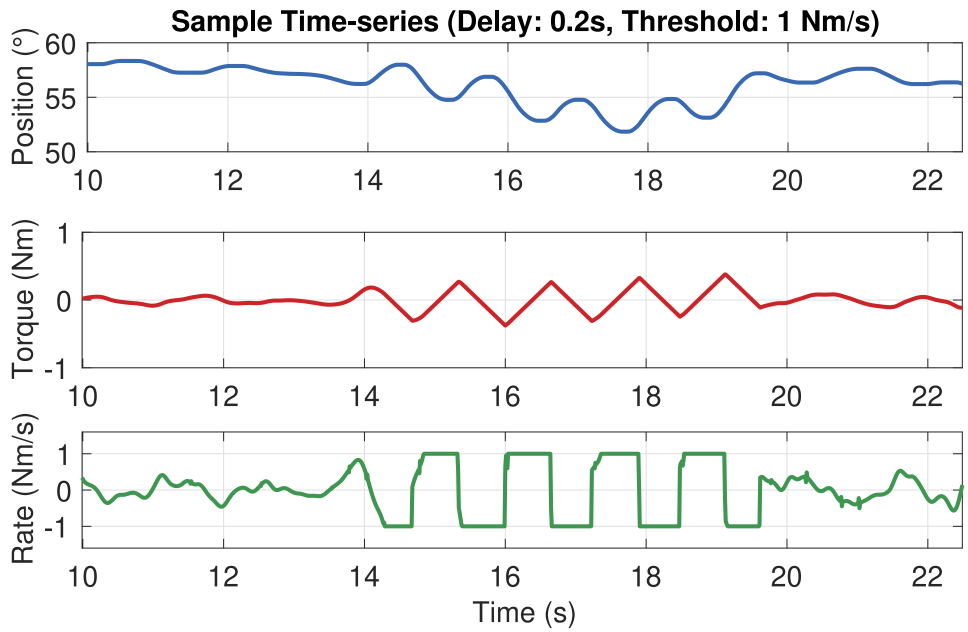


Figure 5.9: A time-series showing the rate-limiter suppressing an unstable oscillation and enabling recovery. Oscillations begin to diverge at 14 s due to high time-delay ($d = 0.2$ s), at which point the behavior follows the predicted limit cycle of Fig. 5.5 until the operator relaxes and regains control at 20 s.

5.6.2.4 Bounded Jerk Perspective

In literature on pHRI with an exoskeleton, it is known that smooth human-like trajectories can be generated by minimizing jerk (time-derivative of acceleration) [122–124]. Consequently, a trajectory with high jerk appears as unnatural and robotic. Thus, another perspective to the method of this study is that it promotes more natural movements by rate-limiting the human-applied torque signals and thus bounding the jerk of the position trajectories generated by the admittance controller.

5.7 Conclusion

In this paper, we show how rate-limiting human-applied torque signals for pHRI on the EXO-UL8 exoskeleton can prevent diverging instability due to time-delay. Specifically, we model the HITL system as: a human as a proportional controller with constant delay, a rate-limiter on the output of the human's applied torques, and a linear second-order virtual dynamics model enabled by admittance control. Our analysis shows that even for large destabilizing time-delays, the trajectories of the HITL system are attracted to a stable limit cycle that can be bound by selecting a sufficiently small rate-limiter threshold. We experimentally validate the rate-limiter through 16 trials corresponding to 4 time-delays and 4 rate-limiter thresholds for each of 10 subjects. Results agree with theory across all time-delays for moderate and large rate-limiter thresholds, and show the limitations of the proportional controller human model for very small rate-limiter thresholds. The results demonstrate that the rate-limiter is effective at preventing instability due to large time-delays, and allows the human to recover from delay-induced oscillations, all with minimal interference to regular operation.

CHAPTER 6

Sensor Reduction

[5] **J. Sun**, Y. Shen, J. Rosen, "Sensor Reduction, Estimation, and Control of an Upper-Limb Exoskeleton," in *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1012-1019, April 2021, doi: 10.1109/LRA.2021.3056366.

6.1 Overview

A multi-DoF exoskeleton relies on an array of sensors to communicate its state (e.g., positions/orientations) and operator-exoskeleton contact interactions (e.g., forces/torques) to its control system. Although sensor redundancy is common in biological systems to cope with uncertainty and partial failure of sensors, in man-made systems, sensor redundancy increases the overall system's cost and control complexity. This study presents a sensor reduction technique for force/torque (F/T) sensors utilizing a Kalman filter-based sensor fusion system in the context of admittance control. The methodology is applied to the EXO-UL8 exoskeleton, which is a powered, redundant, dual-arm, upper-limb robotic system with (7 arm + 1 hand) DoFs incorporating three 6-axis F/T sensors in each arm. Motivated by improving wearability through minimizing human-exoskeleton contact interfaces, which reduces spurious contact forces due to joint misalignment; and reducing cost, the proposed strategy emulates the admittance controller's virtual dynamics with only a subset of sensors, resulting in the physical human-robot interaction feeling the same from the operator's perspective. Experimental results indicate that human-exoskeleton power exchange and actuation stresses of the operator's joints, with the proposed strategy on a subset of two sensors, are comparable to those in the full three-sensor case ($p < 0.01$). The experiments verify the proposed methodology for the EXO-UL8, and support the feasibility of operating other Kalman filter-based sensor fusion systems with fewer sensors without sacrificing transparency in physical human-robot interaction.

6.2 Introduction

In this paper, a method of sensor reduction is presented for a force estimation sensor fusion algorithm in the context of admittance control for the EXO-UL8 exoskeleton [6,58,62].

Force sensing and estimation are prevalent in the field of exoskeleton [28, 52–56] and robotics control [17, 18, 70, 125–128]. Force sensing includes resolving sensor redundancies and finding optimal sensor placement [28, 126–128], whereas force estimation includes sensorless approaches, such as using disturbance observers [28, 55] or other model-based state estimators/filters [17, 18, 54, 56, 125]. These techniques have found applications in teleoperation [17, 18], exoskeleton control [28, 52–56], human-robot interaction [70], and other applications in which the use of sensors is limited by feasibility, reliability, or cost.

Sensorless force estimation, such as in the flying probe of [125], the exoskeletons of [54,55], and the quadrocopter of [70], estimate external contact forces through knowledge of the system dynamics. Unlike the exoskeletons of [28, 55, 56], the EXO-UL8 does not have back-driveable joints in order to achieve higher joint payload capacity. As a result, sensorless approaches could not be utilized.

Applications in which the use of sensors is limited have necessitated the exploration of sensor reduction techniques. These include sensorless approaches, as described above, or reducing the number of required sensors, which is the focus of this research. In the latter case, existing literature has formulated the problem of selecting an optimal subset of sensors as minimizing some cost function [126–128]. While these approaches typically deal with a large number of sensors and are concerned with the optimal subset of sensors, our paper aims to show that different subsets of sensors can be tuned to yield similar dynamic responses as the full set.

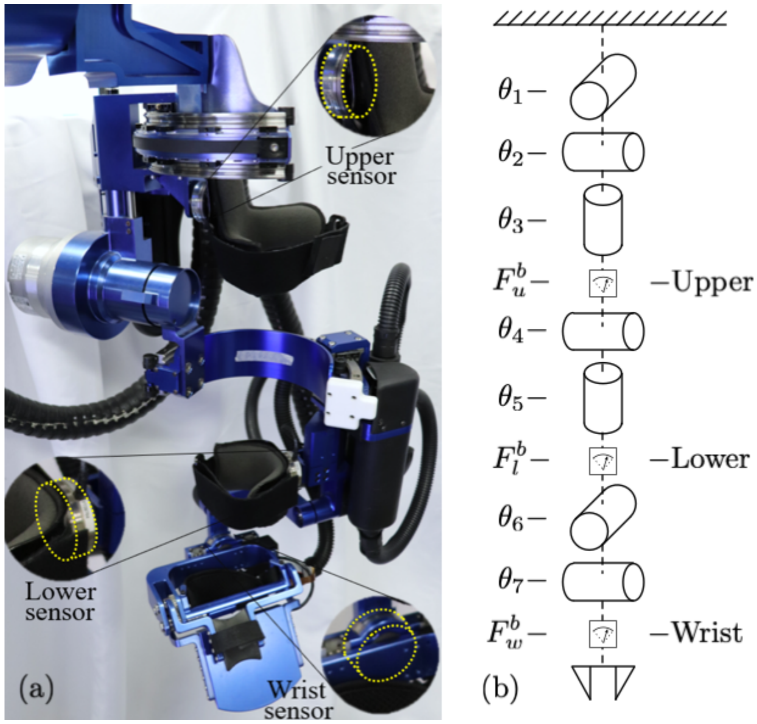


Figure 6.1: (a) On each arm (right arm shown), the upper and lower force/torque sensors interface with an operator's arm via elastic cuff links. The wrist sensor is embedded into the gripper. (b) Each arm can be analyzed as a serial manipulator with joints corresponding to those of a human arm: $\{\theta_1, \theta_2\}$ - shoulder abduction/adduction and flexion/extension, θ_3 - shoulder interior/exterior rotation, θ_4 - elbow flexion/extension, θ_5 - forearm pronation/-supination, θ_6 - wrist extension/flexion, and θ_7 - wrist radial/ulnar deviation.

In resolving sensor redundancy, literature has explored sensor fusion techniques such as Kalman filtering [72,129,130], fuzzy logic approaches [129,131], Monte Carlo methods [132], and other weighted sum approaches [28,58]. Whereas the ARMIN IV+ of [28] uses a constant weighted sum to combine sensor inputs, our approach uses a Kalman filter to account for state-dependency of the sensor fusion gains, as the arm's ability to generate force is position-dependent [62].

Many variations of Kalman filtering are utilized in robotic state estimation and control. [133] implements the Extended Kalman filter to estimate joint angles from the non-linear dynamics of muscle tension control in a redundant musculoskeletal humanoid. [125]

implements a modified Kalman filter with acceleration estimation for a flying probe system. In [134], the authors utilize the Unscented Kalman filter in pose estimation to enable back-stepping control of a mobile robot. In our work, we utilize the linear Kalman filter for sensor fusion.

The EXO-UL8 is a dual-arm, powered, redundant, upper-limb exoskeleton with seven active DoFs and one active gripper DoF on each arm [6, 58, 62] designed to support research efforts in robot-assisted rehabilitation. The exoskeleton tracks an operator's movements through admittance control in joint-space. The admittance controller is driven by operator-applied forces that are measured by three 6-axis force/torque sensors (ATI Mini40) located at the upper arm, lower arm, and wrist, as shown in Fig. 6.1. Reducing the number of required sensors in the EXO-UL8 is motivated by:

1. Improved wearability: During donning, a patient's arm must pass through each of the elastic cuffs, akin to putting one's arm through the sleeve of a sleeved shirt. For patients with neuromuscular disorders such as coupled joint movements or muscular spasticity, such a maneuver is difficult or impossible.

2. Joint alignment: Misalignment of the rotational axes of the EXO-UL8's joints with those of anatomical joints can result in large contact forces to the operator [135]. The absence of a sensor can provide increased scapular movement freedom so that the operator can actively correct for joint misalignment.

3. Reduced cost: If fewer sensors can achieve similar performance, then component cost can be lowered.

While existing literature explores optimal sensor placement, sensor fusion, and sensorless force-estimation and control, the main contribution of our paper is a Kalman filter tuning method to emulate the baseline admittance controller virtual dynamics (based on the full three-sensor case) with only two of the three sensors, resulting in the interaction feeling the same from the operator's perspective.

The rest of the paper is organized as follows: section 6.3 describes the filtering and control strategies of the EXO-UL8, section 6.4 examines the minimum required number of sensors and the compensation for a missing sensor, and section 6.5 describes the experimental validation of the proposed method.

6.3 System Architecture

6.3.1 Cascaded Control Scheme

The EXO-UL8 implements a cascaded control scheme in which the sensor fusion block combines measured forces into a torque signal. The torques are then input to the admittance controller, which generates joint-space trajectories tracked by PD motor joint controllers. Fig. 6.2 shows a block diagram of the control architecture.

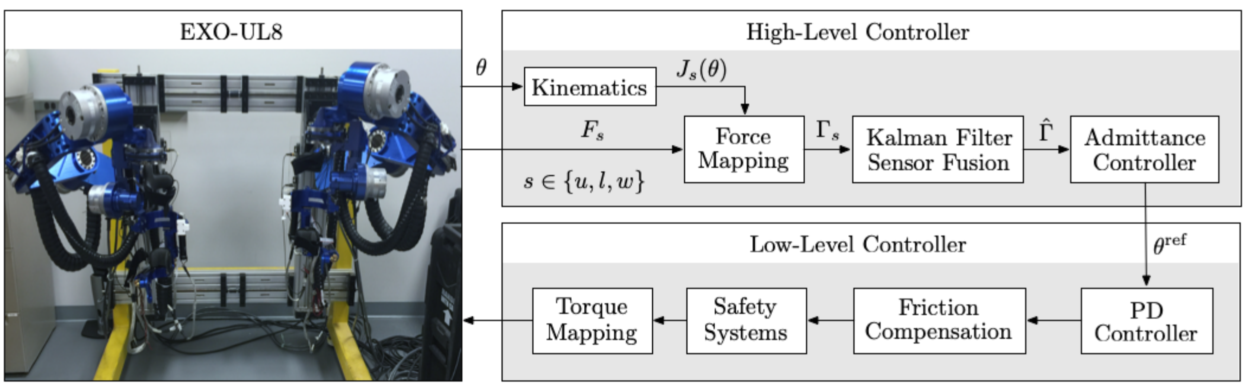


Figure 6.2: The cascaded control scheme of the EXO-UL8 operates at 1kHz and consists of a high-level controller, which contains the Kalman filter-based sensor fusion block and admittance controller, and a low-level controller, which tracks joint-space reference trajectories. Relevant signals are labeled.

6.3.2 Sensor Torque Mapping and Fusion

The EXO-UL8 was originally designed with three 6-axis force/torque sensors on each of its two arms: one at the upper arm (u), one at the lower arm (l), and one integrated into the wrist assembly (w), as shown in Fig. 6.1. Each sensor $s \in \{u, l, w\}$ provides a wrench measurement, $F_s^b \in \mathbb{R}^6$, in its body reference frame, as indicated by the b superscript. To enable compatible operations, each measured wrench F_s^b is transformed to the spatial frame, located at the intersection of the three shoulder axes of rotation, through:

$$F_s^{sp} = \mathrm{Ad}^{\top}_{g_s^{-1}(\theta)} F_s^b, \tag{6.1}$$

where $F_s^{sp} \in \mathbb{R}^6$ expresses the equivalent wrench in the spatial frame, and $\mathrm{Ad}_{g_s} \in \mathbb{R}^{6\times 6}$ is the corresponding adjoint matrix for the homogeneous transformation $g_s \in \mathrm{SE}(3)$ from the spatial frame to the sensor's body frame. The transformed wrenches, F_s^{sp}, are then mapped to joint torques $\Gamma_s \in \mathbb{R}^7$ with the spatial manipulator Jacobian:

$$\Gamma_s = J_s(\theta)^{\top} F_s^{sp}. \tag{6.2}$$

Note that $J_u(\theta) \neq J_l(\theta) \neq J_w(\theta)$ because the dimensions are different due to each sensor being located at a different position along the kinematic chain, as shown in Fig. 6.1.

The torque contributions from the sensors are then combined via a LTI sensor fusion system to yield joint torques $\hat{\Gamma} \in \mathbb{R}^7$ to input to the admittance controller. The sensor fusion system is represented as:

$$\begin{aligned} x_\Gamma[k+1] &= A_\Gamma x_\Gamma[k] + B_\Gamma \operatorname{col}(\Gamma_u[k], \Gamma_l[k], \Gamma_w[k]), \\ \hat{\Gamma}[k] &= C_\Gamma x_\Gamma[k] + D_\Gamma \operatorname{col}(\Gamma_u[k], \Gamma_l[k], \Gamma_w[k]), \end{aligned} \tag{6.3}$$

where $x_\Gamma[k] \in \mathbb{R}^{n_\Gamma}$ is the state of the sensor fusion at time step k, $(A_\Gamma, B_\Gamma, C_\Gamma, D_\Gamma)$ are state-space matrices in minimal realization, and $\operatorname{col}(\cdot, \ldots, \cdot)$ produces a column vector from its arguments. The sensor fusion system is expressed in discrete-time to support software implementation.

6.3.3 Sensor Fusion via Kalman Filtering

A Kalman filter-based sensor fusion combines the torques from the sensors $(\Gamma_u, \Gamma_l, \Gamma_w)$ into a single torque estimate $\hat{\Gamma}$. Since the joint torques are generated from human-applied forces, the exact signal is not known a priori. Therefore, the process equation for Γ is modeled as a random walk, similar to the technique used in [6, 70]:

$$\Gamma[k+1] = \Gamma[k] + (\Delta t) w_\Gamma[k], \tag{6.4}$$

where Δt is the sampling period, and $w_\Gamma[k] \sim \mathcal{N}(0, Q_\Gamma)$, where Q_Γ is an empirically tuned covariance matrix. The torques $\Gamma_u, \Gamma_l, \Gamma_w$ are then treated as measurements with additive Gaussian noise to the Kalman filter:

$$z[k] := \begin{bmatrix} \Gamma_u[k] \\ \Gamma_l[k] \\ \Gamma_w[k] \end{bmatrix} + \begin{bmatrix} w_u[k] \\ w_l[k] \\ w_w[k] \end{bmatrix}, \tag{6.5}$$

$$= \begin{bmatrix} \mathbb{I}_{3\times3} & \mathbb{0}_{3\times4} \\ \mathbb{I}_{5\times5} & \mathbb{0}_{5\times2} \\ \multicolumn{2}{c}{\mathbb{I}_{7\times7}} \end{bmatrix} \Gamma[k] + \begin{bmatrix} w_u[k] \\ w_l[k] \\ w_w[k] \end{bmatrix}, \tag{6.6}$$

$$:= H\Gamma[k] + \mathrm{col}(w_u[k], w_l[k], w_w[k]), \tag{6.7}$$

where $z[k] \in \mathbb{R}^{15}$ is a combined vector of joint torques from the sensors. $w_u[k] \sim \mathcal{N}(\mathbb{0}_{3\times1}, R_u)$, $w_l[k] \sim \mathcal{N}(\mathbb{0}_{5\times1}, R_l)$, and $w_w[k] \sim \mathcal{N}(\mathbb{0}_{7\times1}, R_w)$, where $R_u \in \mathbb{R}^{3\times3}$, $R_l \in \mathbb{R}^{5\times5}$, and $R_w \in \mathbb{R}^{7\times7}$ are the noise covariance matrices corresponding to the upper, lower, and wrist sensor, respectively. Let $\hat{\Gamma} \in \mathbb{R}^7$ be the MMSE estimate of Γ, $P_p \in \mathbb{R}^{7\times7}$ be the variance of the a priori, $P_m \in \mathbb{R}^{7\times7}$ be the variance of the a posteriori, and $R := \mathrm{diag}(R_u, R_l, R_w)$. Then, the update equations for the Kalman filter become:

Initialization:

$$\hat{\Gamma}[0] = \mathbb{0}_{7\times 1}, \tag{6.8}$$

$$P_m[0] = (\Delta t)^2 Q_\Gamma. \tag{6.9}$$

A Priori Update:

$$P_p[k] = P_m[k-1] + (\Delta t)^2 Q_\Gamma. \tag{6.10}$$

A Posteriori Update:

$$K[k] := P_p[k]H^\top (HP_p[k]H^\top + R)^{-1}, \tag{6.11}$$

$$\hat{\Gamma}[k] = (\mathbb{I} - K[k]H)\hat{\Gamma}[k-1] + K[k]z[k], \tag{6.12}$$

$$\begin{aligned} P_m[k] = {} & (\mathbb{I} - K[k]H)P_p[k](\mathbb{I} - K[k]H)^\top \\ & + K[k]RK[k]^\top, \end{aligned} \tag{6.13}$$

where $K[k] \in \mathbb{R}^{7\times 15}$ is defined as the Kalman gain at time step k. Note that equation (6.13) implements the Joseph form for numerical stability.

The Kalman filter implemented in this form is not time-invariant, so it cannot be expressed in the form of equation (6.3). However, this is not problematic because convergence of the Kalman filter is guaranteed by $(\mathbb{I}_{7\times 7}, H)$ being detectable and $(\mathbb{I}_{7\times 7}, Q_\Gamma^{1/2})$ being stabilizable [71], where $\mathbb{I}_{7\times 7}$ is the state transition matrix in equation (6.4). Then, let P_∞ be the steady-state a posteriori variance calculated from the discrete algebraic Riccati equation and let $K_\infty = P_\infty H^\top (HP_\infty H^\top + R)^{-1}$ be the steady-state Kalman gain [72]. The updated equations become:

$$\hat{\Gamma}[k] = (\mathbb{I} - K_\infty H)\hat{\Gamma}[k-1] + K_\infty z[k], \tag{6.14}$$

which is a discrete-time, linear time-invariant system.

6.4 Sensor Reduction

6.4.1 Admittance Controller

The estimated joint torques, $\hat{\Gamma}$, from the sensor fusion system are then used to drive a first-order reference-generation model in joint-space:

$$\tau_j \dot{\theta}_j^{\text{ref}} + \theta_j^{\text{ref}} = a_j \hat{\Gamma}_j,\ j \in \{1, \ldots, 7\}, \tag{6.15}$$

where $\tau_j, a_j \in \mathbb{R}$, $\tau_j > 0$, $a_j > 0$ are the time constant and DC gain of the model for joint j, and $\theta^{\text{ref}} \in \mathbb{R}^7$ is the generated reference signal to be tracked by the motor controllers. These constants are experimentally tuned to achieve responsive behavior of the EXO-UL8, as qualitatively determined by test users. In the Laplace domain, each channel of equation (6.15) has a pole at $s = -\tau_j^{-1}$, which is stable since $\tau_j > 0$. Furthermore, the model can be exactly discretized to:

$$\theta_j^{\text{ref}}[k+1] = e^{-\frac{\Delta t}{\tau_j}} \theta_j^{\text{ref}}[k] + a_j(1 - e^{-\frac{\Delta t}{\tau_j}}) \hat{\Gamma}_j[k], \tag{6.16}$$

for each joint, $j \in \{1, \ldots, 7\}$. The discretized model ensures that discretization errors are minimal.

A summary of the control scheme implementation is given in Procedure 3. The procedure is implemented as an interrupt handler for a timer with interrupt frequency of 1kHz.

6.4.2 Minimum Number of Sensors

In non-singular configurations of the joint angles, the wrist Jacobian, $J_w(\theta)$, is the only Jacobian that can affect all seven dimensions of the joint torque vector, $\hat{\Gamma}$. For this reason, it must be included in the control strategy. Additionally, at least one of the upper or lower sensors must also be present. To illustrate this requirement, consider the case in which only the wrist sensor provides the joint torques used by the admittance controller:

$$\hat{\Gamma} = \Gamma_w = J_w(\theta)^\top F_w. \tag{6.17}$$

Procedure 3 1kHz Timer Interrupt Handler

$\theta \leftarrow$ Read joint angles
for $s \in \{u, l, w\}$ **do**
    $F_s^b \leftarrow$ Read force sensor
    $F_s^{sp} \leftarrow \mathrm{Ad}^{\top}_{g_s^{-1}(\theta)} F_s^b$ ▷ Eqn (6.1)
    $\Gamma_s \leftarrow J_s(\theta)^{\top} F_s^{sp}$ ▷ Eqn (6.2)
KF a priori update   ▷ Eqn (6.10)
$\hat{\Gamma} \leftarrow$ KF a posteriori update   ▷ Eqns (6.11)-(6.13)
for $j \in \{1, \ldots, 7\}$ **do**
    $\theta_j^{\mathrm{ref}} \leftarrow$ Update virtual dynamics   ▷ Eqn (6.16)
Send θ^{ref} to Low-Level Controller

In order for the single sensor to provide enough information to fully control the exoskeleton, the map between the space of wrenches ($\mathbb{R}^6$) to the space of joint torques ($\mathbb{R}^7$) must be surjective. Due to the limited dimensionality of the space of wrenches, there does not exist a mapping that satisfies this requirement. In fact, the wrench can only map to a six-dimensional subspace in $\mathbb{R}^7$, assuming that the Jacobian does not lose rank from the exoskeleton being in a singular configuration. The orthogonal complement of the column space of $J_w(\theta)^{\top}$ is the left nullspace of $J_w(\theta)^{\top}$, or simply the nullspace of $J_w(\theta)$. Since $J_w(\theta) \in \mathbb{R}^{6\times 7}$ and has full row rank, the dimension of its nullspace is one, and corresponds to the manifold of internal motions on which $J_w(\theta)\dot{\theta} = 0$. This manifold contains the motions along the swivel angle in which the wrist maintains its position in end-effector space while the elbow is free to rotate [62, 136]. The redundancy of the EXO-UL8 means that the wrist sensor alone cannot provide enough information, so at least one other sensor must also be present. Thus, a total of two sensors are utilized.

6.4.3 Feasibility of Two Sensors

When two of the 6-axis force/torque sensors are included, a total of twelve inputs are provided to the exoskeleton to actuate seven joints. The Kalman filter in the admittance control scheme serves as a sensor fusion system whose outputs are estimates of the joint torques. Feasibility of requiring only two sensors is equivalent to controllability of the Kalman filter when interpreted as an LTI system. Therefore, if the sensor fusion system described by equation (6.14) is controllable, there exist inputs from the sensors that can drive the torque estimate to any point in the state-space. The pair $(K_\infty, \mathbb{I} - K_\infty H)$ is controllable if and only if its controllabilty matrix is full rank:

$$\mathcal{C} = [K_\infty \quad (\mathbb{I} - K_\infty H)K_\infty \quad \dots \quad (\mathbb{I} - K_\infty H)^6 K_\infty]. \tag{6.18}$$

Since the Kalman filter converges, as shown in subsection 6.3.3, the steady-state Kalman gain, $K_\infty \in \mathbb{R}^{7\times 15}$ is necessarily full rank. The first block column of $\mathcal{C}$ is K_∞, so the controllability matrix must already have a column rank of 7. Therefore, the sensor fusion system is controllable and the inputs from the two sensors are sufficient to produce any joint torque estimate.

6.4.4 Sensor Fusion Tuning to Compensate for Fewer Sensors

The absence of an upper or lower sensor impacts the interaction dynamics experienced by the operator; more force may be required to move the exoskeleton in certain directions. To ensure that the interaction feels the same from the operator's perspective when only two of the three sensors are utilized, the baseline (full three-sensor case) admittance controller virtual dynamics must be emulated. This is achieved by tuning the Kalman filter in either of the reduced-sensor cases to have the same filter dynamics as in the baseline. In both cases, the admittance controller receives the same input and generates the same virtual dynamics. The details of this tuning strategy are explained in this section.

Let the sensor configurations be denoted as:

(A) All three sensors (upper, lower, wrist),

(B) Lower and wrist sensors only,

(C) Upper and wrist sensors only.

From equation (6.15), the same joint trajectories are generated if the $\hat{\Gamma}$ output from the Kalman filter remains the same. Equation (6.14) shows that the steady-state Kalman gain, K_∞, and the measurement matrix, H, directly affect the filter dynamics. For the subsequent analysis, let:

$$H_{lw} := \begin{bmatrix} \mathbb{I}_{5\times 5} & \mathbb{0}_{5\times 2} \\ & \mathbb{I}_{7\times 7} \end{bmatrix}, \tag{6.19}$$

which denotes the measurement matrix used to define $z_{lw}[k]$ in the Kalman filter a posteriori update equations, and corresponds to the case in which the upper sensor is absent (config. B). Then, to ensure that equation (6.14) remains the same in configurations A and B, it is required that:

$$K_\infty H = \tilde{K}_\infty H_{lw}, \tag{6.20}$$

where $\tilde{K}_\infty$ denotes the modified steady-state Kalman gain. Expanding equation (6.20), the requirement becomes:

$$\begin{aligned} &H^\top \left(HP_\infty H^\top + R\right)^{-1} H \\ &= H_{lw}^\top \left(H_{lw}P_\infty H_{lw}^\top + R_{lw}\right)^{-1} H_{lw} \end{aligned} \tag{6.21}$$

where $R_{lw} \in \mathbb{R}^{12\times 12}$ is the new diagonal measurement covariance to be determined. Note that the estimation error covariance, P_∞, should be the same in both cases to ensure that the removal of one sensor does not change the steady-state performance of the Kalman filter. Then, the objective is to solve equation (6.21) for the only unknown, R_{lw}.

Note that the matrices H and H_{lw} are related by:

$$H = \begin{bmatrix} \mathbb{I}_{3\times 3} & \mathbb{O}_{3\times 9} \\ & \mathbb{I}_{12\times 12} \end{bmatrix} H_{lw} := EH_{lw}. \tag{6.22}$$

Finding an appropriate matrix E is always possible when H_{lw} has full row rank, which is a necessary requirement for the Kalman filter to converge in this case. Then, the left side of equation (6.21) becomes:

$$= H_{lw}^{\top} E^{\top} \left(HP_{\infty}H^{\top} + R \right)^{-1} EH_{lw}. \tag{6.23}$$

By equating the matrices between the $H_{lw}^{\top}$ and H_{lw} terms, R_{lw} is solved as:

$$R_{lw} = \left[E^{\top} \left(HP_{\infty}H^{\top} + R \right)^{-1} E \right]^{-1} - H_{lw}P_{\infty}H_{lw}^{\top}. \tag{6.24}$$

A similar analysis calculates $R_{uw} \in \mathbb{R}^{10\times 10}$ for configuration C. Equation (6.24) computes the necessary measurement noise covariance matrix to achieve equal filter dynamics to the nominal case, despite the absence of a sensor.

When the Kalman filter reaches steady state, the contribution of each measurement to the estimate is proportional to the inverse of the associated noise variance. Thus, to visualize how sensor contributions change, it suffices to consider how the noise variances in R_{lw} (config. B) and R_{uw} (config. C) differ from those in R_u, R_l, R_w (config. A). For example, the contribution of the lower sensor to joint 1 in config. A is:

$$\frac{1/R_l(1,1)}{1/R_u(1,1) + 1/R_l(1,1) + 1/R_w(1,1)} \approx 0.256. \tag{6.25}$$

However, when the upper sensor is removed (config. B), the contribution of the lower sensor becomes:

$$\frac{1/R_{lw}(1,1)}{1/R_{lw}(1,1) + 1/R_{lw}(6,6)} \approx 0.767. \tag{6.26}$$

The increase indicates that when the upper sensor is removed, the Kalman filter places greater emphasis on the measurement of the lower sensor in order to yield the same dynamics. Fig. 6.3 summarizes the distributions of sensor contributions in each of the three configurations.

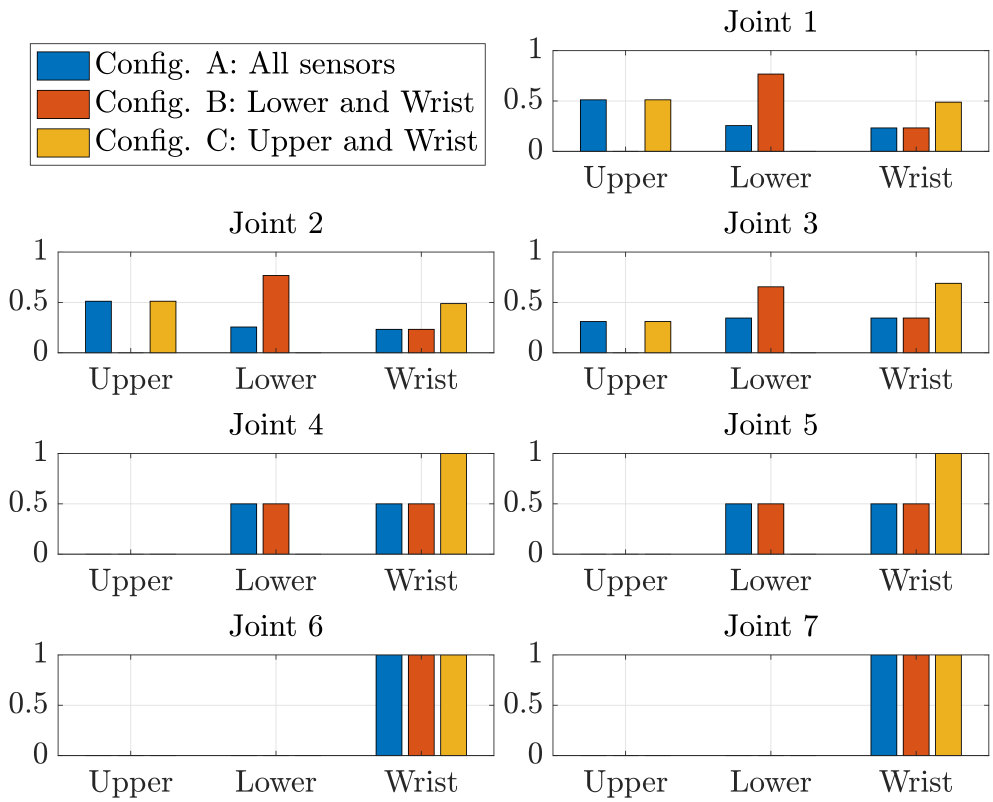


Figure 6.3: The theoretical normalized sensor contributions to the estimated torque are determined by normalizing the reciprocal of the variance values for each sensor with the sum of the reciprocal variance values in each sensor configuration. Joints located farther down the kinematic chain are affected by fewer sensors, as in the case of Joints $4-7$. In all cases, the removal of a sensor redistributes the relative contributions of the remaining sensors.

6.5 Experiments

All experiments in this study were performed with a healthy right-handed participant (male, 25-years-old) following an approved Institutional Review Board protocol (IRB #18-00766).

6.5.1 Performance Metrics

6.5.1.1 NASA Task Load Index (NASA-TLX)

The term *transparency* is a measure of the exoskeleton's tracking performance to an operator's movements. Although it can be quantified using the metrics defined below, a qualitative assessment of ease of control and wearability, as provided by the operator, is also an important indication of performance. To this end, the NASA-TLX survey [137] was utilized to assess the quality of the interaction and ease of donning for each of the three sensor configurations.

6.5.1.2 Power Exchange

In an ideal interaction, no force occurs at the physical human-exoskeleton interface (sensor locations). During motion, this is equivalent to zero mechanical power exchanged. Therefore, the power exchanged through the sensors can quantify the transparency of the interaction; the smaller the power exchanged, the more ideal the interaction. Let $v_s^{sp} \in \mathbb{R}^6$ be the linear and angular velocity of sensor s expressed in the spatial frame. Then the instantaneous power exchange for sensor s is the inner product between the wrench and velocity: $P_s(t) := \langle F_s^{sp}(t), v_s^{sp}(t)\rangle$. The mean power exchange over an interval $t \in [0, T]$ is then:

$$P_s^{\text{avg}} := \frac{1}{T}\int_0^T \langle F_s^{sp}(\tau), v_s^{sp}(\tau)\rangle \,\mathrm{d}\tau. \tag{6.27}$$

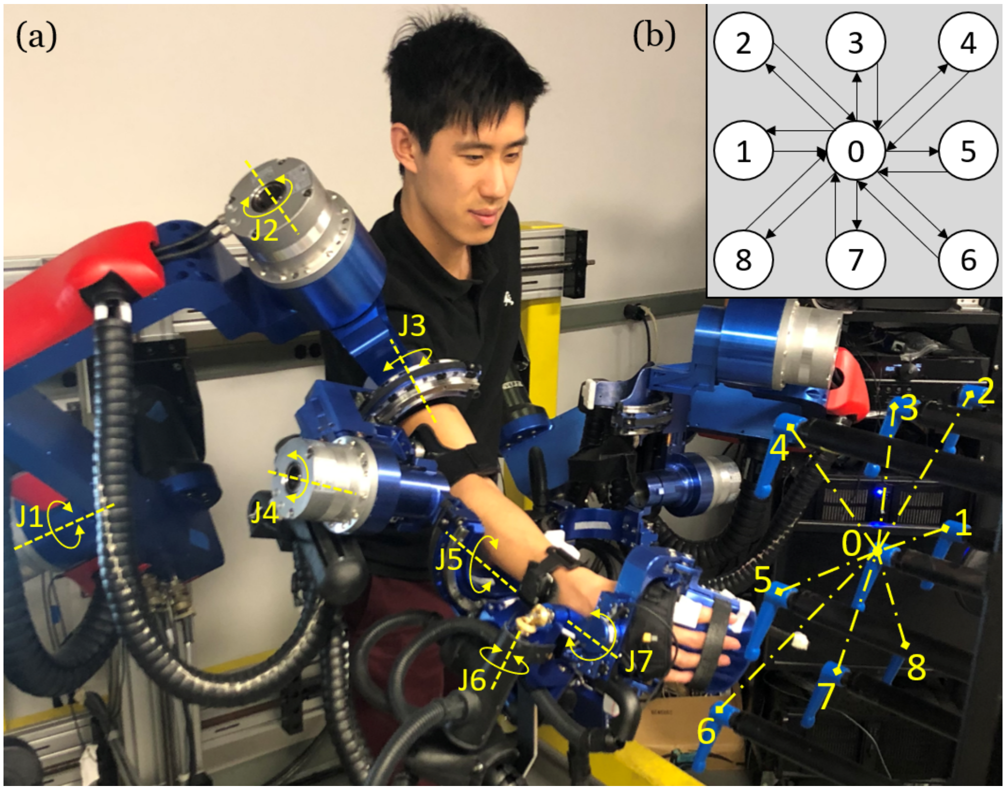


Figure 6.4: (a) A subject wears the exoskeleton to accomplish the trajectory-following tasks; (b) Planned trajectory.

6.5.1.3 Actuation Stress

As another metric for transparency, the actuation stress is defined as a normalization of the effort contributed by each joint in the operator's arm during motion. The less torque each joint has to produce relative to its limit, the lower the actuation stress. Quantitatively, the actuation stress for joint $j \in \{1, \ldots, 7\}$ is defined as:

$$S_j(t) := \frac{|\hat{\Gamma}_j(t)|}{\Gamma_j^{\max}} \times 100\%, \tag{6.28}$$

where $\hat{\Gamma}_j(t)$ is the estimated torque from the Kalman filter, and $\Gamma_j^{\max}$ is the max joint torque that a human arm is able to exert. Table 6.1 shows typical anatomical values for $\Gamma_j^{\max}$ [138]. Note that the torque limits are direction-dependent due to differences in concentric and eccentric muscle contractions.

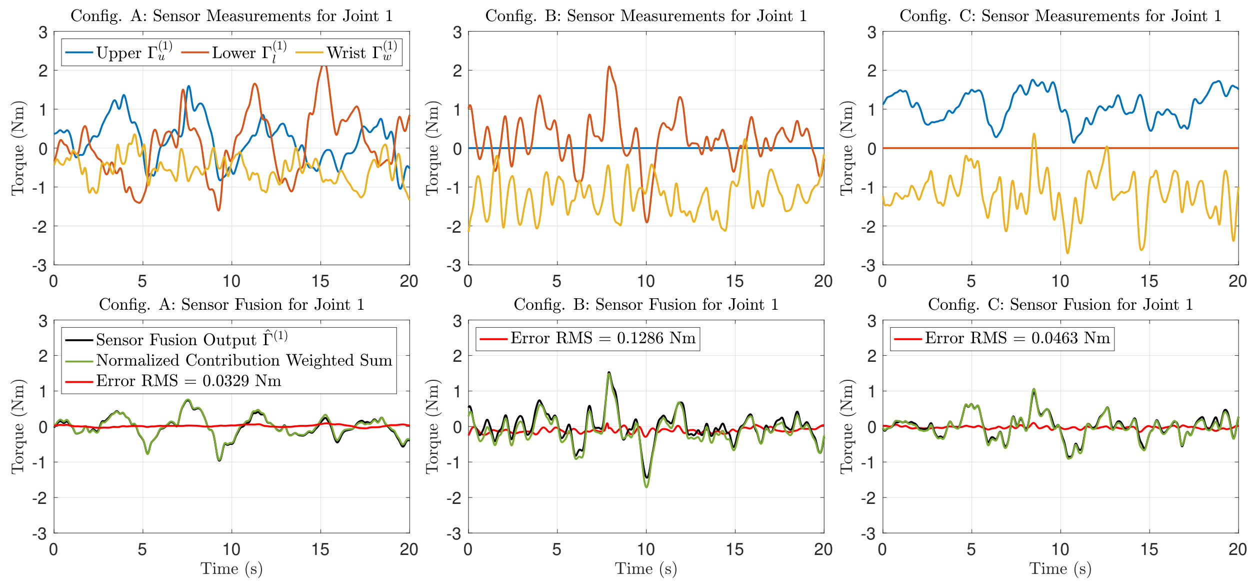


Figure 6.5: A sample 20 second duration data fusion time-series is shown for joint 1 for the three sensor configuration cases. The three plots in the top row show the torque measurements Γ_s from equation (6.2) for the three sensors in the three configurations. The bottom row shows the corresponding sensor fusion outputs $\hat{\Gamma}$. The torque measurements are also combined in a weighted sum with the normalized contribution values from Fig. 6.3. The estimated torques output from the sensor fusion algorithm shows strong agreement with the expected results based on the compensated sensor noise covariance matrix in equation (6.24). The differences of the signals is also shown, and quantified by its RMS value. The small magnitudes of errors indicate that the analysis based on the steady-state Kalman filter in equation (6.14) is valid for the time-varying filter.

6.5.2 Experimental Setup

A reaching trajectory, as shown in Fig. 6.4, is used to assess the three sensor configurations. For configurations B and C, the attachment cuff for the unused sensor was also detached. As operator-exoskeleton force exchange occurs via the attachment cuffs, without a sensor to quantify the interaction forces, local dynamics may not be accurately captured, and may consequently harm transparency. Additionally, one of the primary motivators for removing sensors was to improve wearability by reducing the number of attachment cuffs that an

Table 6.1: Direction-dependent joint torque limits

Joint	Positive Limit (Nm)	Negative Limit (Nm)
Shoulder	Flexion: 13.13	Extension: 8.90
	Adduction: 14.49	Abduction: 15.62
	Internal Rotation: 11.59	External Rotation: 11.63
Elbow	Flexion: 10.75	Extension: 8.76
Wrist	Pronation: 3.39	Supination: 1.42
	Extension: 2.11	Flexion: 1.55
	Radial Deviation: 2.67	Ulnar Deviation: 1.98

operator's arm has to pass through to donn the exoskeleton.

The target trajectory in Cartesian space is designed to exercise a large range of motion. Physical markers (15 cm apart from each other) delineate the trajectory in front of the exoskeleton as shown in Fig. 6.4. The plane of the targets is located 75 cm in front of the operator, at a height at which the operator's outstretched arm is perpendicular to the operator's body when touching the topmost target. A 5 cm rubber pointer at the end-effector is used to make contact with the targets. To ensure comparable timescales across all experimental trials, the subject is given 2 seconds to complete each segment of the trajectory without stopping, for a total of 8×2 (forward and back) segments. A metronome with a 2 second period is used to pace the experiment. The subject also wears short-sleeved clothing to prevent inaccurate sensor readings caused by nonlinear deformation of clothing. Prior to each trial, the subject is given 3 minutes to become familiar with the operation of the exoskeleton. A total of 10 trials for each sensor configuration is carried out to ensure the statistical significance of results.

6.5.3 Results and Discussion

6.5.3.1 Qualitative Assessment

The NASA-TLX assessment for the three sensor configurations is shown in Table 6.2. A lower number is favorable for all metrics except for Performance. The numbers in parentheses for configuration B and configuration C indicate the change from the corresponding task load in configuration A, which serves as the baseline. Qualitative assessment from the subject indicates little change in terms of exoskeleton operation and wearability, which is the desired result. However, configuration B indicates a slight increase in operational difficulty, likely due to more inaccuracies in estimating the torques of the shoulder joints as the closest sensor (upper) is removed in configuration B. This sensing limitation is also evident in the quantitative results described in the subsequent sections.

Table 6.2: NASA-TLX Assessment for each Sensor Configuration

Better	Scale (1 - 20)	Config. A	Config. B	Config. C
↓	Mental Demand	5	5	5
↓	Physical Demand	7	8 (+1)	6 (-1)
↓	Temporal Demand	10	11 (+1)	10
↑	Performance	5	6 (+1)	5
↓	Effort	7	8 (+1)	7
↓	Frustration	5	5	5

6.5.3.2 Sensor Contribution

Fig. 6.5 shows experimental data for the three sensor configurations for joint 1. The top row plots the joint torques converted from the sensor readings (equation (6.5)). The bottom row shows the output of the time-varying Kalman filter and a weighted sum of the torques from the first column using the theoretical contributions in Fig. 6.3. These were plotted together to show strong agreement, which indicates convergence of the Kalman filter. Demonstrating that the time-varying filter achieves expected results with experimental data validates the steady-state Kalman filter assumption used in calculating the theoretical sensor contributions of Fig. 6.3.

The bottom row of Fig. 6.5 also shows the error and its RMS to quantify the disagreement between the expected filter output and measured filter output. While configuration C shows agreement to the baseline (config. A) in terms of error RMS, configuration B shows a larger error, which agrees with the qualitative assessment. This may be caused by the removal of the closest sensor to joint 1 (upper sensor in configuration B). The lower and wrist sensors are located farther along the kinematic chain than the upper sensor, so their accurate estimation of the torque on joint 1, as compared to that of the upper sensor, is affected by a greater number of intermediate joints.

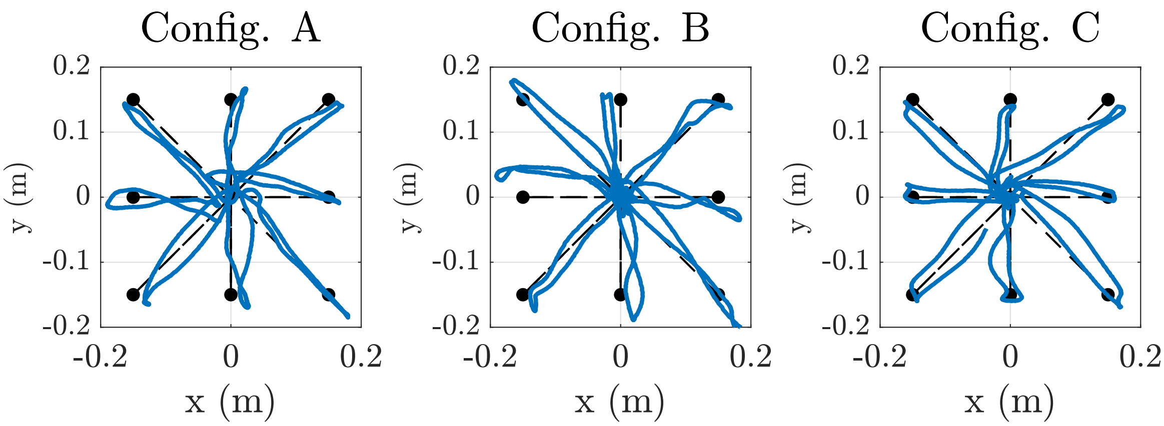


Figure 6.6: End-effector trajectories in the plane of the target pattern for the three sensor configurations. Trajectories are overlayed onto the target pattern shown in Fig. 6.4. Each of the three configurations allows for satisfactory performance in enabling the operator to follow the target pattern.

6.5.3.3 Power Exchange

Sample end-effector trajectories are shown in Fig. 6.6. Mean power exchange for the trials are computed with equation (6.27) and represented by the box-and-whisker plots in Fig. 6.7. Experimental results show that the compensated Kalman filters resulted in lower power exchange as compared to the uncompensated cases ($p < 0.01$). Statistical significance of the power exchange results was evaluated using the two-sample t-test. The null hypothesis for each sensor in configurations B and C was that the power exchange distributions of the compensated and uncompensated cases had equal mean but unknown variance. The alternative hypothesis was that the distributions had unequal means. In all four cases (B - lower, B - wrist, C - upper, C - wrist), the p-values were less than 0.01, with the largest being $p = 0.0089$ for the wrist sensor in configuration C, indicating that re-tuning the Kalman filter was statistically significant in improving transparency, when measured with the power exchange metric.

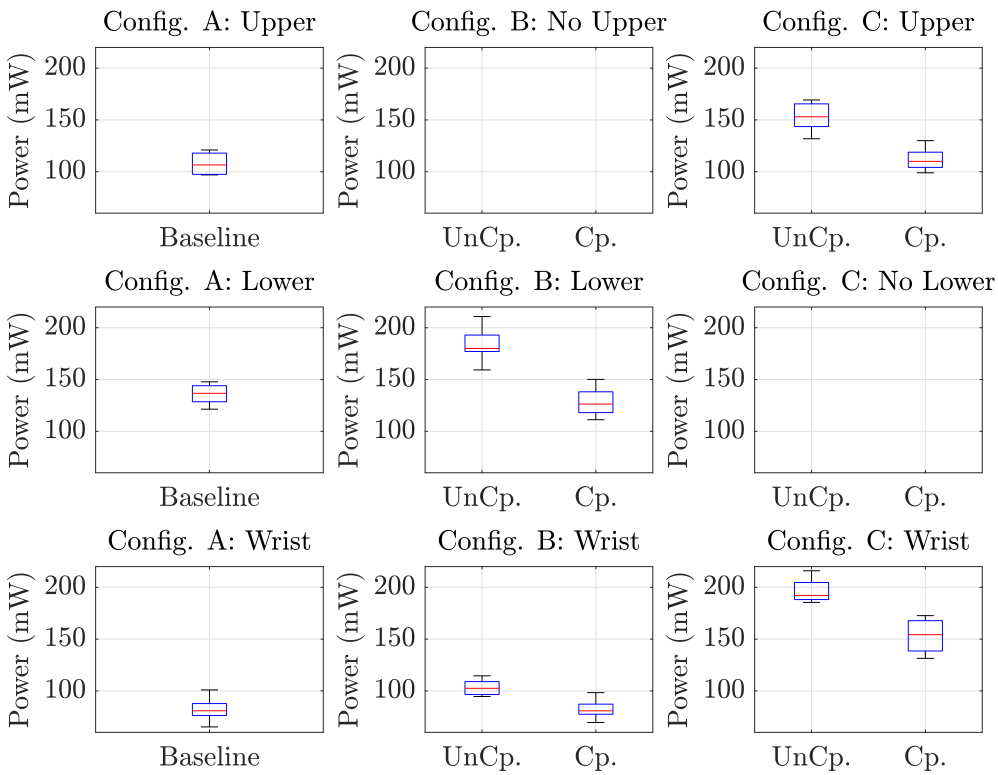


Figure 6.7: The columns show the power exchange for configurations A, B, and C (left to right). Uncompensated (Uncp.) refers to the sensor configuration applied but without re-tuning the Kalman filter; i.e., the filter operates under the assumption that all sensors are present, even though a sensor is physically removed. On the other hand, compensated (Cp.) refers to tuning the filter's noise covariance matrices according to equation (6.24). Results indicate that after tuning, average power exchange decreases ($p < 0.01$), indicative of more transparent human-exoskeleton interaction.

In configuration B, the mean power exchange of the compensated case closely matched that of the baseline, albeit with more variance. This is likely caused by the same limitation evident in the NASA-TLX qualitative assessment of Table. 6.2 and error RMS of Fig. 6.5: the removal of the upper sensor places more emphasis on the lower sensor to estimate torques for the shoulder joints (1-3), which may introduce additional uncertainties as there are now more intermediate joints between the shoulder and its closest sensor (lower).

In configuration C, the power exchange in the compensated case is higher than in the baseline. Since the lower and wrist sensors are only 12.5 cm apart, the absence of the lower sensor and its attachment cuff may cause the full mass of the operator's forearm to rest on only the wrist attachment, resulting in higher sensor readings. This anomaly is not present in configuration B (upper sensor absent) because the mass of the operator's upper arm is supported by their shoulder and does not rest on the upper sensor. With the upper sensor absent, the compensated cases match more closely with the baseline.

6.5.3.4 Actuation Stress

The actuation stresses are computed with equation (6.28), averaged across the trials, and shown in Fig. 6.8. Between the compensated and uncompensated cases, all joints except for joint 2 show a lower actuation stress when the remaining sensors are re-tuned according to section 6.4.4, which agrees with the power exchange results. The discrepancy for configuration C is likely caused by the redistribution of the forearm's mass as previously described in the Power Exchange subsection. In both configurations, the compensated cases show closer agreement with the baseline configuration in which all sensors are present.

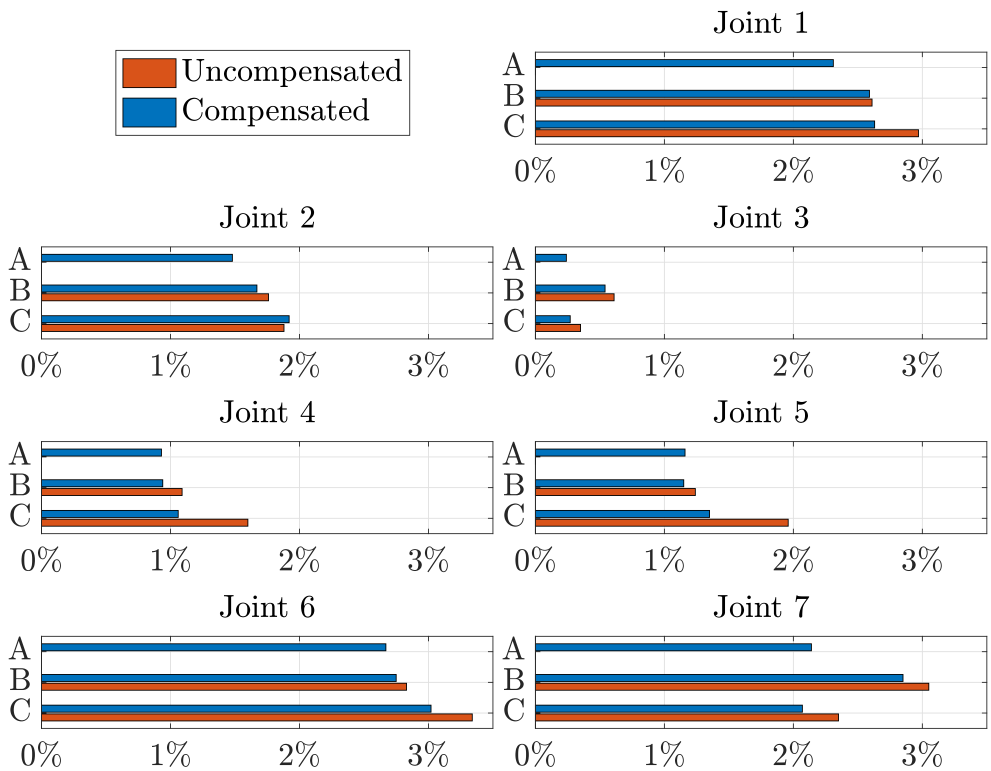


Figure 6.8: Actuation stresses for the three sensor configurations show that compensation (re-tuning the Kalman filter) results in closer values to the baseline (config. A).

6.6 Conclusion

This study proposed a tuning method for removing sensors in a Kalman filter-based sensor fusion system in which any reasonable subset of sensors yields the same filter dynamics as with the full set of sensors. The dynamical impact of operating with a subset of sensors without tuning was demonstrated experimentally, which motivates the need for a systematic tuning strategy. The proposed method was verified on the EXO-UL8 exoskeleton where the output of the Kalman filter drove an admittance controller. The tuning method was applied to two different sensor configurations (configs. B and C), and retained similar performance as the original full set of sensors (config. A). Experiments performed with the EXO-UL8 quantified actual performance by calculating operator-exoskeleton power exchange and actuation stress. Results agree with theoretical expectations and support the feasibility and utility of the method.

A limitation of the method arises when sensors are located kinematically far from the joints whose torque are being estimated, such as with joint 1 in configuration B. Qualitative and quantitative assessments indicate a decrease in operator-exoskeleton transparency due to inaccuracies introduced by more intermediate joints. This limitation may be further studied by quantifying transparency as a function of sensor placement, and then implementing the optimal placement.

The proposed sensor reduction method could be applied to any physical system that implements a Kalman filter-based sensor fusion strategy, which is pervasive in the field of robotics. For future work, applying the tuning method to other robotic devices and systems using heterogeneous sensors would broaden the utility of the method. Specifically in the context of the EXO-UL8, further work may be done to explore sensor reduction in bimanual operation, or comparison to other force sensing strategies in the literature.

CHAPTER 7

Constrained Admittance Control

[1] **J. Sun**, Y. Foroutani, J. Rosen, "Virtually Constrained Admittance Control using Feedback Linearization for Physical Human-Robot Interaction with Rehabilitation Exoskeletons." *Under review.*

7.1 Overview

Robot-assisted rehabilitation focuses in part on path-based assist-as-needed reaching rehabilitation, which dynamically adapts the level of robot assistance during physical therapy to ensure patient progress along a predefined trajectory without fostering overreliance on the system. Additionally, bimanual exoskeletons have enabled asymmetric rehabilitation schemes, which aim to promote motor recovery by leveraging the patient's healthy side to guide the rehabilitation through interactions with objects in virtual reality that replicate activities of daily living. Within the context of physical human-robot interaction implemented with admittance control, these tasks can be formulated as constraints on the space of allowable motions. This study introduces a feedback linearization-inspired time-invariant controller that enforces these motion constraints by isolating the component of the admittance control dynamics transversal to the constraint, and then implementing a stabilizing force field. The methodology is applied to two rehabilitation tasks: (1) a path-guided reaching task with restoring force field, and (2) a bimanual interaction with a virtual object. Each task is then evaluated on one of two drastically different exoskeleton systems, namely: (1) the V-Rex, a non-anthropomorphic full-body haptic device, and (2) the EXO-UL8, an anthropomorphic bimanual upper-limb exoskeleton. The two systems exist on opposite ends of the task/joint space control: non-redundant/redundant, off-the-shelf (industrial)/custom, non-anthropomorphic/anthropomorphic spectra. Experimental results validate and support the methodology as a generalizable approach to enabling constrained admittance control for rehabilitation robots.

7.2 Introduction

The field of robot-assisted rehabilitation within the context of pHRI has experienced a surge in attention in recent years [139–142] in areas focused on improving rehabilitation effectiveness and safety. One such instance is AAN, which adapts the level of assistance provided by the robot during physical therapy to ensure patient progress without fostering overreliance on the system. Furthermore, the development of bimanual exoskeletons has enabled asymmetric rehabilitation schemes, which aim to promote motor control recovery by leveraging the patient's healthy side to guide the rehabilitation through VR-based training tasks that replicate ADLs [143,144]. These VR-based studies have introduced dynamic and motivating therapy experiences that adapt to individual participants, drawing upon principles of flow theory [145] and utilizing reinforcement learning [142] to maximize effectiveness. The developments showcase the importance of pHRI in robot rehabilitation, but also expose new challenges in control that must be addressed before its widespread adoption can be realized.

In AAN robot-assisted rehabilitation, virtual force fields are often used to assist patients in following predefined paths, as part of reaching-based rehabilitation. The restoring force is often generated by virtual spring [146, 147] or spring-damper [148] systems. However, such systems can generate large forces if they start far from the path, making them potentially dangerous for therapeutic applications. Thus, other systems have been designed to saturate the restoring force at a desired radius around the desired path using exponential equations [13, 149, 150]. Velocity-field trajectories, where the path is defined as a velocity profile and does not directly depend on the current position of the robot, have also been explored in conjunction with adjustable deadzones and adaptive learning strategies [14,151].

However, a major limitation of these approaches is that the path's geometry has to be considered in the force-restoring system, which complicates their implementation and limits their generalizability.

As most ADLs are bimanual [144,152], robot-assisted rehabilitation has also delved into bimanual rehabilitation strategies, primarily facilitated through bimanual exoskeletons [5, 57,139,144,153], and often synergistically integrated with immersive VR environments [140, 141,154]. These training schemes can involve haptic interaction with virtual objects, which is known to be an important aspect of the motor learning process when used in conjunction with visual feedback [120,121]. However, a challenge of creating stable interaction is the rendering of stiff objects, which require large forces to be generated from small displacements [144,155]. Complex geometries also complicate computing interaction forces, which may require convex approximations. These challenges have spurred the exploration of position-based techniques [156], or more advanced simulation tools [66,157].

In either application, the nominal virtual dynamics of the rehabilitation robot are restricted to a subset. In the case of reaching rehabilitation, a stabilizing controller is used to limit the motion along a predefined path. On the other hand, bimanual interaction with a virtual rigid object restricts the overall space of motions to the subset in which the relative poses of the hands are constant. Constraining pHRI motion to a subset has been explored through various methods. Guidance virtual fixtures produce restoring forces to keep the desired motion along a predefined virtual path [101,158–160]. Similarly, other works utilize a virtual viscoelastic coupling to attract the robot's motion to a surface [161]. These techniques do not consider the system dynamics, which can affect the guidance performance. Transverse feedback linearization, which decouples the system dynamics into tangential and transversal components [162,163], has found application in path-constrained pHRI [7]. While the decoupled dynamics can simplify the design of feedback controllers, such techniques may require local coordinates for both components, which may exist for paths due to their one dimensional nature, but can be difficult to find in general. Hybrid force-position control has also been employed to restrict motions to surfaces for sanding robots [164,165]. However, the additional complexity of impedance control at the surface may not be necessary for applications that are more concerned about spatial accuracy.

In this paper, we present and experimentally validate a novel feedback linearization-inspired admittance control methodology for enabling virtual holonomically constrained pHRI admittance control in robot rehabilitation applications. Specifically, we model the motion constraint as a submanifold formed by the zero level set of a twice-differentiable function, before applying ideas from feedback linearization to isolate the component of the nominal virtual dynamics transversal to the submanifold, without requiring local coordinates for the tangential component. A time-invariant controller then stabilizes this transversal subsystem, while being parameterized by a constraint strength parameter that governs how stringently the constraint is enforced. The methodology is then experimentally validated on two tasks common in robot-rehabilitation: (1) a path-guided reaching task with a virtual restoring force field in AAN rehabilitation schemes, performed on the V-Rex full-body haptic exoskeleton, and (2) a bimanual virtual object interaction found in VR-based asymmetric bimanual rehabilitation, performed on the EXO-UL8 exoskeleton. Demonstrating the applicability of the proposed method on two markedly different rehabilitation tasks, on contrasting exoskeleton systems - existing on opposite ends of the task/joint space control, non-redundant/redundant, off-the-shelf (industrial)/custom, non-anthropomorphic/anthropomorphic spectra - illustrates the method's generalizability and versatility as a novel way to enable constrained admittance control in robot rehabilitation applications.

7.3 Methodology

Our proposed approach requires an admittance control system, whose virtual dynamics represent the desired unconstrained pHRI motion. The constraint is then modeled as the zero level set of some smooth constraint function that only depends on the virtual position. We introduce a parameter $\gamma \in [0, 1] \subset \mathbb{R}$, called the *constraint strength*, which parameterizes how stringently the constraint should be enforced; a value of ($\gamma = 0$) corresponds to unconstrained motion, whereas ($\gamma = 1$) fully constrains the motion. The max value represents a perfectly stiff constraint, which is necessary for emulating stable interaction with non-deformable virtual objects - typical in VR-based rehabilitation environments. Any intermediate value of γ allows for some violation of the constraint set that is also resisted by some restoring force, representative of typical behavior in path-following AAN schemes. Thus, the desired behaviors in different applications of robot-assisted rehabilitation can be selected by varying γ, which is summarized below:

$$\gamma \in \begin{cases} \{0\}, & \text{Unconstrained (nominal)}, \\ (0,1), & \text{Partially constrained}, \\ \{1\}, & \text{Fully constrained}. \end{cases} \tag{7.1}$$

7.3.1 Virtual Dynamics

Admittance control is commonly used for pHRI, in which human-applied forces, either measured or estimated, are used to propagate virtual dynamics, which are typically second-order and represent mass-damper systems [4, 67–69]. The trajectories of these dynamics are then used as reference signals for the robot, which, assuming its controller is sufficiently performant, appears to move like the virtual dynamics. By assuming satisfactory tracking performance, only the virtual dynamics are considered from this point onward. Let $\theta(t) \in \mathbb{R}^n$ represent the generalized virtual position and $\tau(t) \in \mathbb{R}^n$ be the human-applied force, where n is the number of DoFs. For each DoF, the dynamics can be parameterized by a virtual mass $m_i \in \mathbb{R}_{>0}$ and damping $b_i \in \mathbb{R}_{\geq 0}$:

$$m_i\ddot{\theta}_i(t) + b_i\dot{\theta}_i(t) = u_i(t), \tag{7.2}$$

where $i \in \{1, \dots, n\}$, $u \in \mathbb{R}^n$ is the input to the virtual system, and $u(t) = \tau(t)$. The explicit dependency on time notation will be dropped for conciseness. The virtual second-order dynamics can include more complexity by treating each virtual link as a rigid body, resulting in the dynamics:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + N(\theta) = u, \tag{7.3}$$

where $M(\theta) \in \mathbb{R}^{n\times n}$ is a positive-definite inertia matrix, $C(\theta, \dot{\theta}) \in \mathbb{R}^n$ represents the Coriolis and centripetal terms, and $N(\theta) \in \mathbb{R}^n$ is the gravity vector. Note that equation (7.2) can be written in the form of equation (7.3) by letting $M(\theta) = \mathrm{diag}(m_1, \ldots, m_n)$, $C(\theta, \dot{\theta}) = \mathrm{col}(b_1\dot{\theta}_1, \ldots, b_n\dot{\theta}_n)$, and $N(\theta) = 0_{n\times 1}$, where diag constructs a diagonal matrix from its arguments and col stacks its arguments vertically. To write the dynamics more concisely, define $x := (\theta, \dot{\theta}) \in \mathbb{R}^{2n}$ to be the state, and rewrite the dynamics as:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} \dot{\theta} \\ -M(\theta)^{-1}[C(\theta, \dot{\theta}) + N(\theta)] \end{bmatrix} + \begin{bmatrix} 0_{n\times n} \\ -M(\theta)^{-1} \end{bmatrix} u, \\ &:= f(x) + g(x)u, \end{aligned} \tag{7.4}$$

and define $f : \mathbb{R}^{2n} \to \mathbb{R}^n$ and $g : \mathbb{R}^{2n} \to \mathbb{R}^{n\times n}$ accordingly. Equation (7.4) represents the general form of the virtual second-order dynamics, which will be used for all subsequent sections.

7.3.2 Constraint Set Definition

To represent the constraint, let $h : \mathbb{R}^{2n} \to \mathbb{R}^k$ be a twice continuously differentiable function in the state, henceforth referred to as the constraint function, and satisfying $1 \leq k \leq n$ and $0_{k\times 1}$ being a regular value. Since only the class of holonomic constraints are considered, let the constraints be modeled as equality constraints on the state using $h(x) = h(x_{1:n}) = 0_{k\times 1}$, where the $1 : n$ subscript refers to the first n coordinates of x. Next, define the constraint set to be the submanifold formed by the zero level set of h:

$$\Omega := \{x \in \mathbb{R}^{2n} \mid h(x) = 0_{k\times 1}\}. \tag{7.5}$$

Then, as long as the state can be restricted to Ω, the constraint equation is satisfied.

7.3.3 Objective Formulation

To formalize the constrained motion requirements, a time-invariant feedback control law $u(x, \gamma)$ should be designed to satisfy the following three objectives:

Objective 1 (Transparency): The dynamics should be unconstrained ($u = \tau$) for trajectories inside the constraint set. Moreover, when $\gamma = 0$, this requirement should hold over the entire domain $\mathbb{R}^{2n}$.

Objective 2 (Attractiveness): Unforced trajectories ($\tau = 0$) with initial conditions outside the constraint set should move towards it; i.e., $\inf_{x' \in \Omega} \|x(t) - x'\| \to 0$ as $t \to \infty$ for any $\gamma > 0$. However, when $\gamma = 1$, even forced trajectories ($\tau \neq 0$) should be attracted.

Objective 3 (Invariance): When the dynamics are fully constrained ($\gamma = 1$), any trajectory that enters the constraint set should remain there regardless of human-applied forces, τ; i.e., if $x(0) \in \Omega$, then $x(t) \in \Omega$ for all $t \geq 0$.

7.3.4 Virtual Constraints Controller

This subsection utilizes feedback linearization in order to decouple the virtual dynamics into components that are tangential and transversal to the constraint manifold. Then, a stabilizing control law is implemented on the transversal component, while human-applied forces are projected onto the tangential component. To this end, define a virtual output $y = h(x)$ so that stabilizing Ω becomes output regulation of y. Then, the dynamics of y can be determined by differentiating it with respect to time:

$$y = h(x), \tag{7.6}$$

$$\dot{y} = \frac{\partial h}{\partial x}\dot{x} = \begin{bmatrix} \frac{\partial h}{\partial \theta} & 0_{k \times n} \end{bmatrix} \dot{x} = \mathcal{L}_f\, h(x), \tag{7.7}$$

$$\ddot{y} = \mathcal{L}_f^2\, h(x) + \mathcal{L}_g\, \mathcal{L}_f\, h(x) u, \tag{7.8}$$

where $\mathcal{L}_f$ and $\mathcal{L}_g$ are the Lie operators along vector fields f and g, respectively. The dynamics of y result in a vector relative degree of $(2, \ldots, 2)$ due to the constraints being holonomic and the virtual dynamics of equation (7.4) being second-order, allowing the system with virtual output y to be input-output feedback linearizable [7,162,163,166]. It is also assumed that the choice of h results in $\mathcal{L}_g \mathcal{L}_f h(x) \in \mathbb{R}^{k\times n}$ having linearly independent rows on $\Omega^c = \mathbb{R}^{2n} \setminus \Omega$, where the set is assumed to be nonempty. Then, for each $i \in \{1, \ldots, k\}$, define the coordinate transformation:

$$\eta^i(x) := \begin{bmatrix} h_i(x) \\ \mathcal{L}_f h_i(x) \end{bmatrix}, \tag{7.9}$$

so that its dynamics are linear for some virtual control input, $v_i = \mathcal{L}_f^2 h_i(x) + \mathcal{L}_g \mathcal{L}_f h_i(x)u$:

$$\dot{\eta}^i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \eta^i + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_i. \tag{7.10}$$

Each linear system η_i represents the dynamics of a component of y, so y can be regulated by stabilizing each η_i subsystem with a suitable state feedback:

$$v_i = -k_i \eta^i, \tag{7.11}$$

for some $k_i \succ 0_{1\times 2}$, where $\succ$ refers to element-wise inequality. Although a linear controller suffices, any controller $v_i(\eta_i)$ that stabilizes the origin of equation (7.10) will stabilize Ω. Note that $\eta := \mathrm{col}(\eta^1, \ldots, \eta^k) : \mathbb{R}^{2n} \to \mathbb{R}^{2k}$ is not a diffeomorphism unless $k = n$. When $k \neq n$, motion is allowed on the $n - k$ dimensional space Ω. To determine the input u to the admittance controller, the virtual input $v := \mathrm{col}(v_1, \ldots, v_k)$ that stabilizes Ω needs to be transformed back into x-coordinates.

When $\gamma = 1$, the human-applied forces τ should not interfere with the stability of Ω, so its component in the subspace spanned by the rows of $\mathcal{L}_g \mathcal{L}_f h$ needs to be replaced by $v := \mathrm{col}(v_i, \ldots, v_k)$. The remaining component of τ respects the constraint and can be computed by projecting τ into the space tangential to the constraint manifold, which is the nullspace of $\mathcal{L}_g \mathcal{L}_f h$. Then, by using the Moore-Penrose inverse of $\mathcal{L}_g \mathcal{L}_f h$, the control input is:

$$\begin{aligned} u = \tau &- \gamma(\mathcal{L}_g \mathcal{L}_f h)^\dagger (\mathcal{L}_g \mathcal{L}_f h)\tau \\ &+ \mathrm{sgn}(\gamma)(\mathcal{L}_g \mathcal{L}_f h)^\dagger (v - \mathcal{L}_f^2 h), \end{aligned} \tag{7.12}$$

where sgn is the signum function defined as:

$$\operatorname{sgn}(x) := \begin{cases} -1, & \text{if } x \leq 0, \\ 0, & \text{if } x = 0, \\ 1, & \text{if } x \geq 0. \end{cases} \tag{7.13}$$

The control law consists of three components:

1. τ, the nominal human-applied forces,
2. $\gamma(\mathcal{L}_g\,\mathcal{L}_f\,h)^{\dagger}(\mathcal{L}_g\,\mathcal{L}_f\,h)\tau$, the γ-scaled component of τ in the subspace transversal to Ω,
3. $\operatorname{sgn}(\gamma)(\mathcal{L}_g\,\mathcal{L}_f\,h)^{\dagger}(v - \mathcal{L}_f^2\,h)$, the virtual transversal stabilizing controller, transformed to x-coordinates.

The inclusion of γ in two of the terms allows for adjusting the motion constraint behavior using a single parameter. The proposed control law leads to the following theorem.

Theorem 1. *The control law of equation (7.12) satisfies the three main objectives: (1) transparency, (2) attractiveness, and (3) invariance.*

Proof of Theorem 1. To satisfy the first objective, consider a trajectory $x(t) \in \Omega$. Then, since $h(x)$ is identically zero for this trajectory, the matrix $\mathcal{L}_g\,\mathcal{L}_f\,h(x)$ and its Moore-Penrose pseudoinverse are also zero, of dimension $k \times n$ and $n \times k$, respectively. Thus, substituting these matrices into the control law of equation (7.12) reduces it to $u = \tau$, which is independent of γ, as required. Furthermore, when $\gamma = 0$, the control law also reduces to $u = \tau$. These two cases together satisfy **Objective 1**.

To show that the last two objectives can by satisfied using the control law of equation (7.12), consider the error dynamics away from Ω. Let the error $e \in \mathbb{R}^n$ be defined as

$e := h(x)$, so that its closed-loop dynamics are:

$$\dot{e} = \mathcal{L}_f\, h, \tag{7.14}$$

$$\begin{aligned} \ddot{e} = {} & \mathcal{L}_f^2\, h + (\mathcal{L}_g\, \mathcal{L}_f\, h)\tau \\ & - \gamma(\mathcal{L}_g\, \mathcal{L}_f\, h)(\mathcal{L}_g\, \mathcal{L}_f\, h)^\dagger(\mathcal{L}_g\, \mathcal{L}_f\, h)\tau \\ & + \operatorname{sgn}(\gamma)(\mathcal{L}_g\, \mathcal{L}_f\, h)(\mathcal{L}_g\, \mathcal{L}_f\, h)^\dagger(v - \mathcal{L}_f^2\, h), \end{aligned} \tag{7.15}$$

which substitutes equation (7.12) into the virtual dynamics of equation (7.4). By the assumption that $\mathcal{L}_g\, \mathcal{L}_f\, h$ has linearly independent rows, and is rank k, $(\mathcal{L}_g\, \mathcal{L}_f\, h)(\mathcal{L}_g\, \mathcal{L}_f\, h)^\dagger = I_{k\times k}$. Thus, equation (7.15) reduces to:

$$\ddot{e} = \mathcal{L}_f^2\, h + (1-\gamma)(\mathcal{L}_g\, \mathcal{L}_f\, h)\tau + \operatorname{sgn}(\gamma)(v - \mathcal{L}_f^2\, h). \tag{7.16}$$

Considering the last two objectives, there are two cases of interest: (1) $\gamma > 0$ and the dynamics are unforced ($\tau = 0$), and (2) $\gamma = 1$ and τ can be any value. The first and second case correspond to partially and fully constraining the dynamics to Ω, respectively. However, in either case, equation (7.16) simplifies to:

$$\ddot{e} = v. \tag{7.17}$$

By defining $\epsilon := \operatorname{col}(e_1, \dot{e}_1, \ldots, e_k, \dot{e}_k)$, the error dynamics become:

$$\dot{\epsilon} = \left(I_{k\times k} \otimes \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \right) \epsilon + \left(I_{k\times k} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) v, \tag{7.18}$$

where $\otimes$ is the Kronecker product. Then, substitute in the virtual control law of equation (7.11) by observing that $\epsilon = \eta$. The closed-loop error dynamics of equation (7.18) then becomes block-diagonal:

$$\dot{\epsilon} = \operatorname{diag}\left(\begin{bmatrix} 0 & 1 \\ -k_{1,1} & -k_{1,2} \end{bmatrix}, \ldots, \begin{bmatrix} 0 & 1 \\ -k_{k,1} & -k_{k,2} \end{bmatrix} \right) \epsilon, \tag{7.19}$$

where the matrix is formed as the block diagonal of all the 2×2 blocks. Due to its structure, its eigenvalues are the union of the eigenvalues of each 2×2 block. By the Routh-Hurwitz criteria, each block's eigenvalues are in the open left half-plane if $k_i \succ 0_{1\times 2}$ for all $i \in \{1, \ldots, k\}$. Thus, the error dynamics can be stabilized by an appropriate choice of gains k_i. It is worth noting that although the η-dynamics are linear, the choice of a virtual transversal controller v does not have to be linear. The feedback linearization just allowed the system dynamics of equation (7.4) to be decoupled into transversal and tangential subsystems, so the choice of a stabilizing transversal controller is flexible. In any case, the stability of ϵ holds everywhere in the domain, which satisfies **Objective 2** and **Objective 3**. ■

The implementation of the virtual constraints controller within the admittance controller is shown in Fig. 7.1.

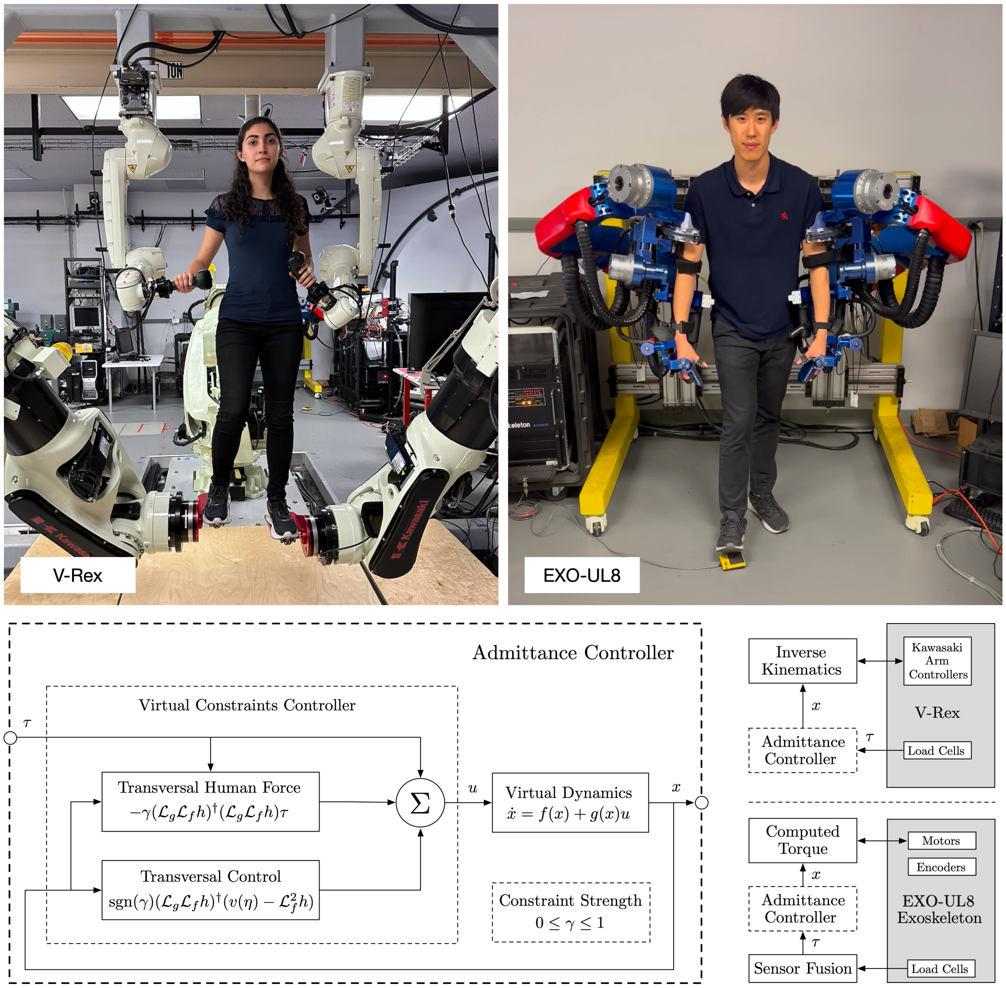


Figure 7.1: (Left) The V-Rex is a full-body haptic system consisting of five robotic serial manipulators interacting with the operator in task-space through the hands, feet, and a harness attached at the pelvis. (Right) The EXO-UL8 is a bimanual upper-limb exoskeleton interacting with the operator in joint-space through three force/torque sensors per arm. (Bottom) The admittance controller consists of the virtual constraints controller of equation (7.12) and the virtual dynamics of equation (7.4). From the nominal human-applied force, τ, the virtual constraints control law replaces a γ-scaled component in the transversal direction with the output of a transversal stabilizing controller. The value of γ determines how strongly the operator can push back against constraint. The same admittance controller structure is used in both the V-Rex and EXO-UL8 exoskeletons, albeit with different virtual dynamics and constraint functions.

7.4 Experimental Setup

In order to demonstrate the versatility of the proposed method, two experiments are conducted: a path-based reaching task with restoring force field, and a bimanual manipulation of a virtual rigid object, each on a different exoskeleton system. A visual depiction of the experiments is shown in Fig. 7.2. For each experiment, the pertinent exoskeleton system is first introduced in the setup, and then the constraint associated with the task is modeled.

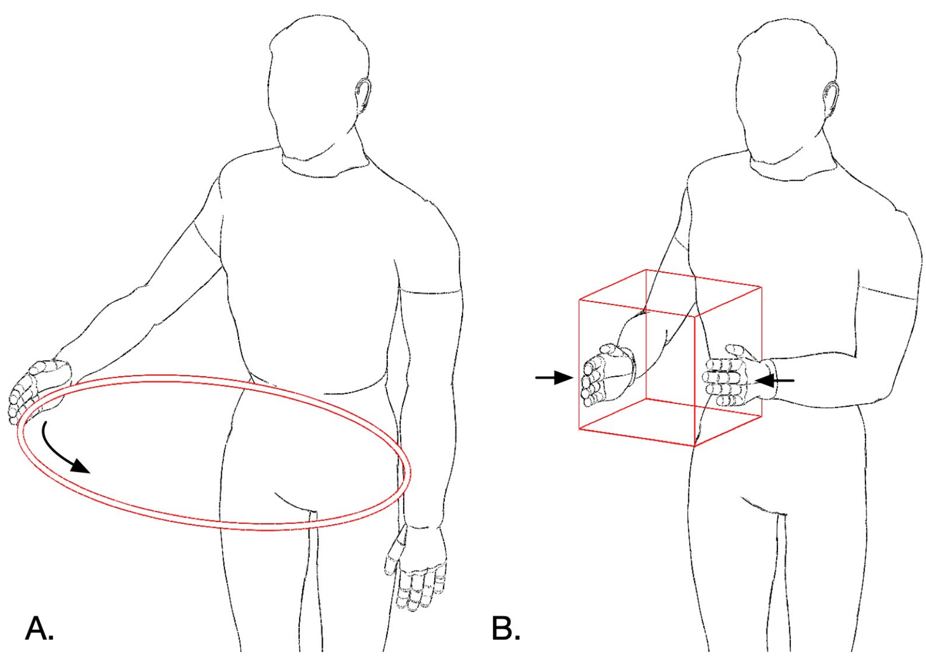


Figure 7.2: The two robot rehabilitation tasks in this study are: (A) a unilateral reaching task constrained to an elliptical path by an adjustable restoring force field, typically found in AAN-based reaching rehabilitation, and (B) a bimanual manipulation task of a virtual rigid object, commonly found in VR-based robot rehabilitation. For each task, the submanifold of allowable motions is modeled as a smooth function's level set, and then used by the virtual constraints controller to restrict nominal motions to the submanifold. The tasks are experimentally verified on the V-Rex full-body haptic exoskeleton and the EXO-UL8 upper-limb bimanual exoskeleton.

7.4.1 Path-Based Assistance-As-Needed

7.4.1.1 V-Rex Full-Body Haptic Exoskeleton

The Virtual Reality Exoskeleton (V-Rex) is a non-anthropomorphic full-body haptic system consisting of five Kawasaki industrial serial manipulators. Two RS-007L manipulators interact with the operator through the hands, two BX-100S manipulators are connected with safety breakaways at the feet, and one CX-210L provides gravity offloading through a harness attached at the pelvis. Each of the five manipulators has six powered revolute DoFs, is independently controlled in task-space, and is also equipped with a six-DoF force/torque sensor at the end effector. On each iteration of the control loop, for each manipulator, the force/torque measurement at the end effector is used to propagate a virtual three-dimensional model of the form in equation (7.2). The virtual state is then transformed into joint-space using an inverse kinematics solver, before serving as reference signals for the embedded Kawasaki arm controller. Fig. 7.1 shows the V-Rex and a block diagram of its control.

7.4.1.2 Elliptical Path Constraint with Restoring Force Field

A path constraint can be implemented with varying levels of constraint strength, ranging from free motion ($\gamma = 0$) to fully constrained ($\gamma = 1$). To illustrate how γ affects the interaction forces and corresponding trajectories during path-based reaching tasks, a constraint set Ω in the form of an ellipse is implemented, while a different ellipse oriented $90°$ from the first is used as a target trajectory. Both ellipses are situated in the XY plane, parallel with the ground. The operator attempts to move along the target trajectory, even though the virtual constraints controller is pushing them onto the constraint set. Although the interaction is through the hand, the setup serves to demonstrate the methodology and can be modified in future experiments to simulate human gait. Equation (7.20) shows the elliptical path constraint, with x_{1c} and x_{2c} defined as the ellipse center, and the minor and major axes,

$a = 0.15$m and $b = 0.2$m, chosen to fit within the V-Rex's workspace:

$$h(x) = \begin{bmatrix} (x_1 - x_{1c})^2/a^2 + (x_2 - x_{2c})^2/b^2 - 1 \\ x_3 \end{bmatrix}. \tag{7.20}$$

The target ellipse has parameters $a = 0.2$m and $b = 0.15$m and is drawn on a table below the manipulator, serving as a visual aid for the operator, as show in Fig. 7.3. Constraint strengths of $\gamma \in \{0.0, 0.3, 0.7, 1.0\}$ are used for the trials.

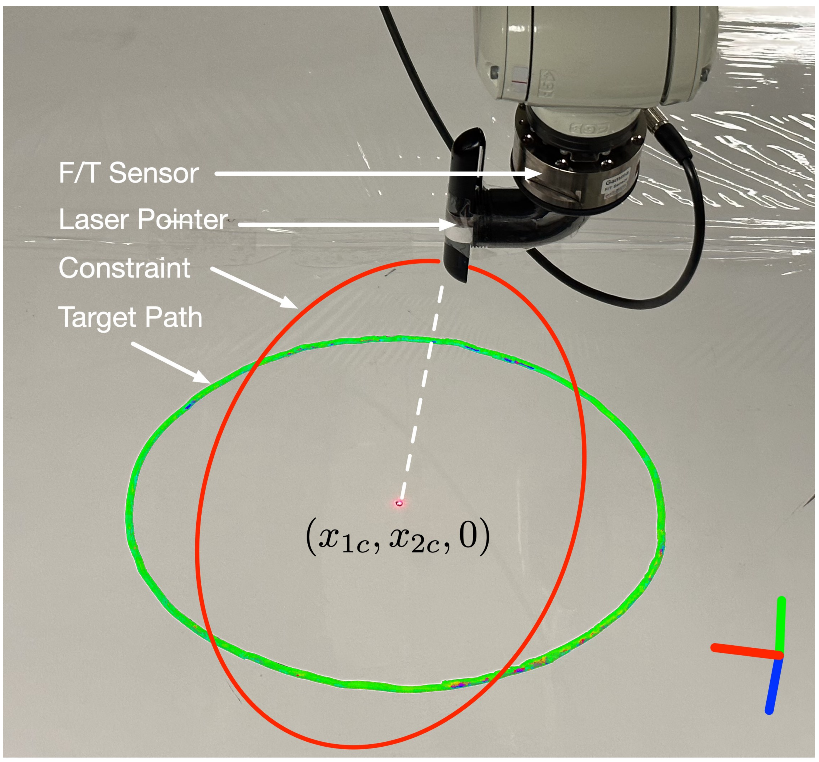


Figure 7.3: The AAN path experiment uses the upper-left arm of the V-Rex constrained to an ellipse (red) with varying constraint strengths, parameterized by γ. The operator attempts to follow the target path (green), which is also an ellipse but rotated 90° about the z-axis in order to illustrate the effect of γ. A laser pointer mounted at the end effector helps to visualize the 2D -projection of the motion onto the plane containing the two ellipses.

The path constraint is enforced through a force field generated by a virtual spring damper system, similar to existing work [148]. This choice of controller does not limit the system as any other stabilizing controller can be used, depending on the desired performance. Furthermore, the experiment aims to demonstrate the main advantage, which is that the transversal component of the constraint is isolated, allowing stabilizing force fields to be developed independently of the path's geometry.

7.4.2 Bimanual Interaction with Virtual Object

7.4.2.1 EXO-UL8 Bimanual Upper-Limb Exoskeleton

The EXO-UL8 is a custom bimanual powered redundant upper-limb anthropomorphic exoskeleton consisting of two arms, each with seven revolute DoFs [4–6, 57]. Each of its two arms is equipped with three six-DoFs force/torque sensors located at the upper arm, lower arm, and wrist. The measured force/torque signals are first fused together using the method from [5], before propagating the virtual dynamics of the admittance control in joint space, which is 14-dimensional and consists of 14 independent second-order systems of the form of equation (7.2), in order to allow each joint to have its own virtual inertia and damping parameters. The combined virtual state is then used as a reference signal that a computed torque controller [40] tracks. Fig. 7.1 shows the system and a block diagram of its control architecture.

7.4.2.2 Relative Pose Constraint

Bimanual interaction with virtual rigid objects involves constraining the overall 14-dimensional space of motions to the subset in which the relative pose between the end effectors is constant. To formulate this constraint, let $T_l, T_r \in \mathrm{SE}(3)$ be the homogeneous transformation from the base (inertial) frame to the left and right end effectors, respectively. Using the *product-of-exponentials* formulation, each transformation can be written as a map of the joint angles:

$$T_l(\theta^l) = \left(\prod_{i=1}^{7} e^{\hat{\xi}_i^l \theta_i^l}\right) T_l(0), \tag{7.21}$$

where $\{\xi_i^l\}_{i=1}^7$ are the twists in local coordinates associated with each of the left joints, and $T_l(0)$ is the transformation in the default configuration. Similarly for the right arm, T_r is a function of θ^r, and parameterized by the right twists $\{\xi_i^r\}_{i=1}^7$. Then, the relative transformation between the end effectors is $T_{rl}(\theta^r, \theta^l) = T_r^{-1}(\theta^r)T_l(\theta^l)$. Using local coordinates for SE(3), and defining $\theta := (\theta^r, \theta^l)$, the constraint function is:

$$h(\theta, \dot{\theta}) = [p(\theta), \alpha(\theta)] - (p_0, \alpha_0), \tag{7.22}$$

where $(p(\cdot), \alpha(\cdot)) : \mathbb{R}^{14} \to \mathbb{R}^3 \times \mathbb{R}^3$ are local coordinates for $T_{rl}(\theta)$, and $(p_0, \alpha_0) \in \mathbb{R}^3 \times \mathbb{R}^3$ is the desired constant relative pose between the end effectors. The coordinates of the desired relative pose, (p_0, α_0), is measured when contact is first made with the virtual object. Note that the choice of local coordinates on SE(3) does not matter as long as they exist over the range of the desired motion. In our experiments, canonical coordinates are used for position and ZYX Tait-Bryan angles for orientation. Specifically, given matrix coordinates for a relative pose, $T_{rl} \in \text{SE(3)}$:

$$T_{rl}(\theta) = \begin{bmatrix} R(\theta) & p(\theta) \\ 0 & 1 \end{bmatrix}, \tag{7.23}$$

$$\alpha(\theta) = \begin{bmatrix} \sin^{-1}(-R_{21}(\theta)/\sqrt{1 - R_{31}(\theta)^2}) \\ \sin^{-1}(-R_{31}(\theta)) \\ \sin^{-1}(-R_{32}(\theta)/\sqrt{1 - R_{31}(\theta)^2}) \end{bmatrix}, \tag{7.24}$$

where the coordinate chart is defined on the subset of SE(3) in which $R_{31} \neq \pm 1$.

The use of a high-dimensional motion set with a constraint manifold $\Omega = h^{-1}(0)$ for which local coordinates on Ω are hard to find aims to demonstrate the generalizability of the proposed feedback linearization-based approach.

7.4.3 Software Implementation and Numerical Considerations

While computing the gradient and Hessians for h in the elliptical path constraint of equation (7.20) may be tractable, the same quantities for the relative pose constraint of equation (7.22) require significantly more work. To ensure correctness and minimize the impact of numerical errors, symbolic tools, such as SymForce [167] or Sympy [65], can symbolically differentiate the quantities and generate corresponding C++ code that can be optimized and evaluated quickly online.

Note that when $x \in \Omega$, $h(x) = 0$, so the matrix $\mathcal{L}_g \mathcal{L}_f h(x)$ and its Moore-Penrose inverse are also zero, of dimension $k \times n$ and $n \times k$, respectively. In practice, when floating-point arithmetic is used, care should be taken to ensure that numerically small values in $\mathcal{L}_g \mathcal{L}_f h(x)$ are treated as zero. Specifically, each row of $\mathcal{L}_g \mathcal{L}_f h(x)$ should first be checked for whether it is numerically non-zero; call the set of row indices $NZ(x) \subseteq \{1, \ldots, k\}$. Next, form the matrix $L = \mathrm{row}(r_1, \ldots, r_k)$, where row r_i is defined as:

$$r_i := \begin{cases} \text{row } i \text{ of } \mathcal{L}_g \mathcal{L}_f h(x), & \text{if } i \in NZ(x), \\ 0_{1\times n}, & \text{otherwise.} \end{cases} \tag{7.25}$$

Then, any instance of the vector $(\mathcal{L}_g \mathcal{L}_f h(x))^\dagger w$, where $w \in \mathbb{R}^k$, such as in the case of equation (7.12), can be written as a linear combination of non-zero columns of $L^\dagger$:

$$(\mathcal{L}_g \mathcal{L}_f h(x))^\dagger w \approx \sum_{i \in NZ(x)} (L^\dagger)_i w_i, \tag{7.26}$$

where $(L^\dagger)_i$ is the i^{th} column of $L^\dagger$.

7.5 Results

7.5.1 Path-Based Assistance-As-Needed

The path-based reaching task assesses the controller with four constraint strengths, ranging from free motion ($\gamma = 0$) to fully constrained ($\gamma = 1$). The operator starts at the left-most edge of the target ellipse (-200mm, -450mm), and the virtual constraints controller is activated. Fig. 7.4 shows the transversal states, defined in equation (7.20), of the manipulator's virtual dynamics as a function of time. In the fully constrained case, the states converge to zero, regardless of the human-applied forces as expected. In the semi-constrained trials, the

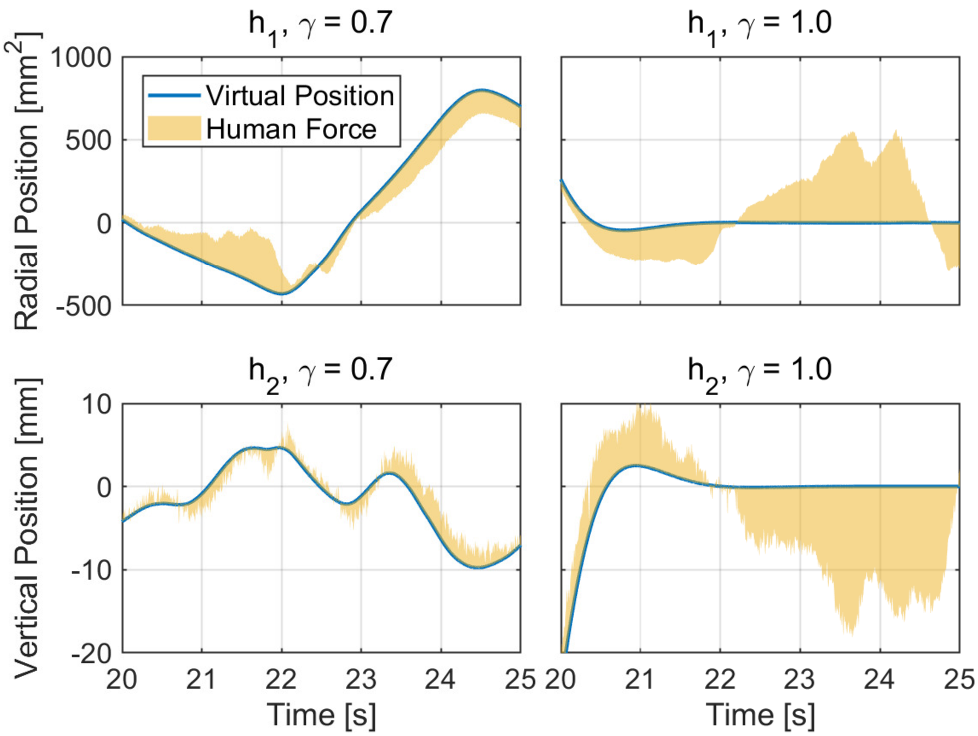


Figure 7.4: The transversal states of the path-constrained reaching experiment, and the corresponding human-applied forces, are shown as timeseries for two different values of constraint strength: $\gamma = 0.7$ (left), and $\gamma = 1.0$ (right). Human-applied forces together with the output of the virtual constraints controller influence the virtual position when $\gamma = 0.7$, allowing the operator deviate from the constraint while feeling a restoring force. However, when $\gamma = 1.0$, the virtual position stabilizes independently of any human-applied force.

human-applied forces can resist the controller and prevent the state from reaching zero, as shown in the left column of Fig. 7.4 and the middle subfigures of Fig. 7.5. In these cases, the operator haptically experiences the restoring force field generated by the controller.

Fig. 7.5 plots the motion trajectories, with either the transversal component of the human-applied forces or the virtual constraint controller's output overlaid. During free motion ($\gamma = 0$), there are no constraint-stabilizing forces. In the partially constrained cases ($\gamma = 0.3, \gamma = 0.7$), the operator can guide the robot along the target path; however, the virtual constraints controller partially resists the motion, resulting in transversal forces

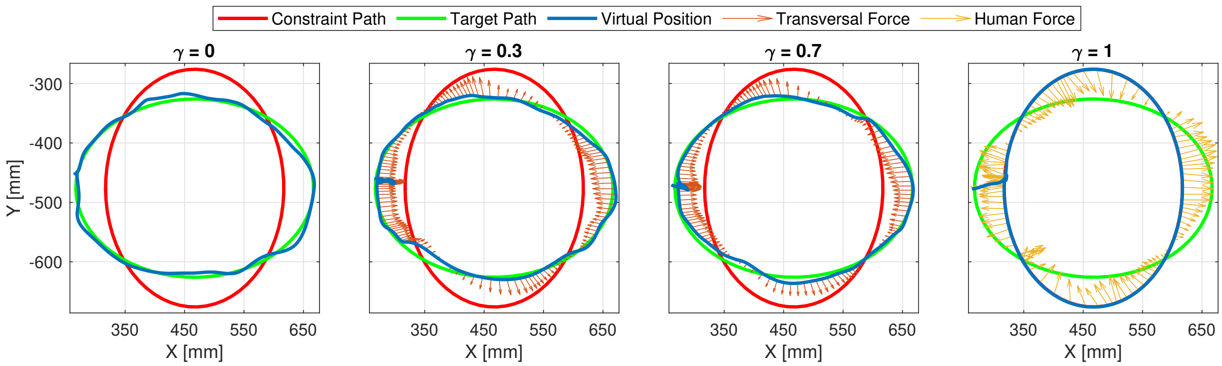


Figure 7.5: The manipulator's trajectories (blue line) and relevant transversal forces are plotted for each constraint strength parameter (γ) in the path-based reaching rehabilitation task. The operator attempts to follow a target trajectory (green line), which is oriented 90° from the constraint path (red line) to demonstrate the impact of γ. In the unconstrained trial ($\gamma = 0$), no restoring forces are present, as expected. When the motion is partially constrained ($0 < \gamma < 1$), the restoring force field (red arrows) generated by the virtual constraints controller push the motion towards the constraint set. In the fully constrained ($\gamma = 1$), the manipulator's trajectory aligns with the constraint set, as expected, despite human-applied forces (yellow arrows) pushing towards the target path. Each trial took approximately 30 seconds to complete.

pushing towards the constraint. In the fully constrained case ($\gamma = 1$), the robot's trajectories remain on the constraint path, despite the human-applied forces pushing towards the target path.

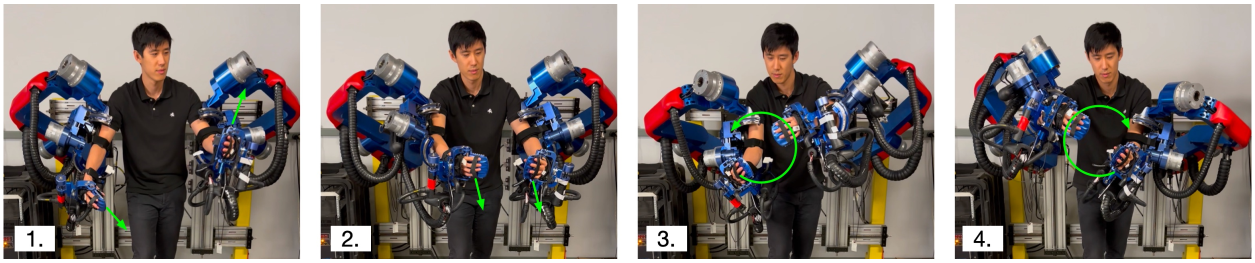


Figure 7.6: Subimages show keyframes of the bimanual interaction with a virtual object. The object is not explicitly defined by its geometry, but rather as a constraint on the total joint space in which the relative pose between the hands is fixed. During keyframe (1), the hands are unconstrained ($\gamma = 0$). In keyframe (2), the constraint is activated by setting $\gamma = 1$. Subsequent motions (2)-(4) show that the virtual rigid coupling between the hands gives the illusion of interacting with a virtual rigid object that can be freely manipulated.

7.5.2 Bimanual Interaction with Virtual Object

The bimanual interaction starts with the operator wearing the EXO-UL8 and freely moving each arm, as shown in subfigure 1 of Fig. 7.6. Once the operator's hands are in a desired configuration, e.g., around the boundary of some virtual object, the current relative pose between the hands is measured and stored as (p_0, α_0), which is used in the constraint function of equation (7.22). The virtual constraints controller is then activated by setting the constraint strength $\gamma = 1$, as shown on at the left gray boundary in Fig. 7.7. Subsequent motion within the constraint manifold physically appear as the hands being rigidly coupled, as seen in subfigures $2-4$ of Fig. 7.6, giving the illusion of interacting with a virtual rigid object.

Furthermore, during this time the component of the human interaction force that is transversal to the constraint manifold is completely rejected, as shown by the human forces in Fig. 7.7. The magnitude and direction of these transversal forces do not impact the constraint controller's tracking performance at all, which is expected from setting $\gamma = 1$. The remaining tangential component of the interaction force component allows the virtual object (a relative pose between the end effectors) to move only on the $14 - 6$ dimensional joint space constraint manifold, as illustrated by subfigures $2 - 4$ of Fig. 7.6. Once the interaction is complete, the constraints are deactivated by setting $\gamma = 0$ at the right gray boundary in Fig. 7.7, and the hands can move independently again.

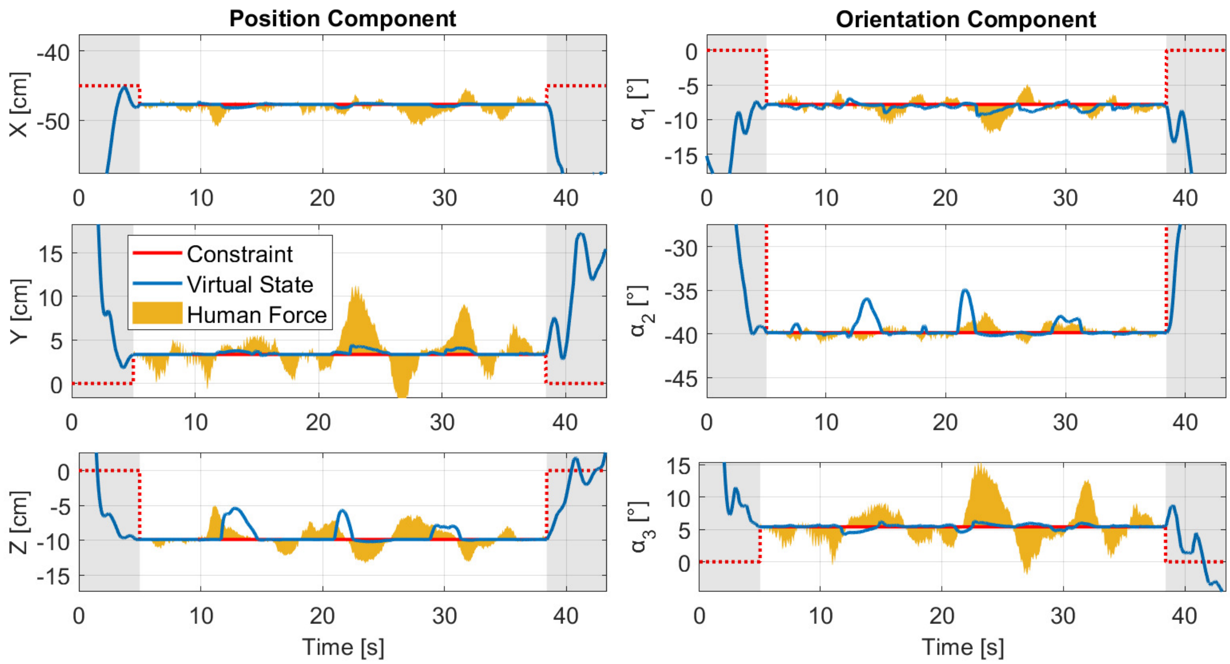


Figure 7.7: Subplots show the six components of the transversal state of the relative pose between the EXO-UL8's end effectors. The gray regions show unconstrained motion ($\gamma = 0$), during which the constraint values are not set (dashed red lines), whereas the region in the middle is fully constrained ($\gamma = 1$). Since the desired pose is snapshot at the start of the constrained motion (left edge of gray region), the virtual state (blue) is equal to the constraint value (red) at this point in time. During the constrained motion, the virtual constraints controller tracks the constraint while rejecting any human-applied forces in the transversal direction (orange) that try to move the virtual state away from the constraint manifold. Note that the spikes in the virtual state are explained in Fig. 7.8.

7.6 Discussion

7.6.1 Path-Based Assistance-As-Needed

The experiment utilized a virtual spring damper system to generate the corrective force-field transversal to the constraint path in order to match existing experiments in the literature. However, the proposed method differs in that the force field is completely agnostic of the constraint path's geometry, and can be generated by any stabilizing controller. Furthermore, while the experiment utilizes a closed elliptical path for demonstrating the path constraint, which is commonly found in AAN rehabilitation tasks, the proposed method is not limited to a specific path or robotic system. Unlike many existing control schemes for path following that require either a parameterized path or one comprised of timed waypoints, this method is time-invariant and can be applied to any valid path without requiring a parameterization for the submanifold, and any admittance control rehabilitation robot utilizing second-order virtual dynamics.

7.6.2 Bimanual Interaction with Virtual Object

In the Z-position and α_2-orientation subplots of Fig. 7.7, the virtual state appears to deviate from the constraint manifold at 13s, 22s, and 30s. This behavior is not a limitation of the virtual constraints controller, but rather due to the hard joint limits of the EXO-UL8. At these instances, joint 6 of the right arm reaches its minimum allowed value of $-30°$ as shown in Fig. 7.8 Since it cannot go lower, the corresponding virtual states h cannot maintain its set points of $h_3 = -10$ cm and $h_5 = -40°$.

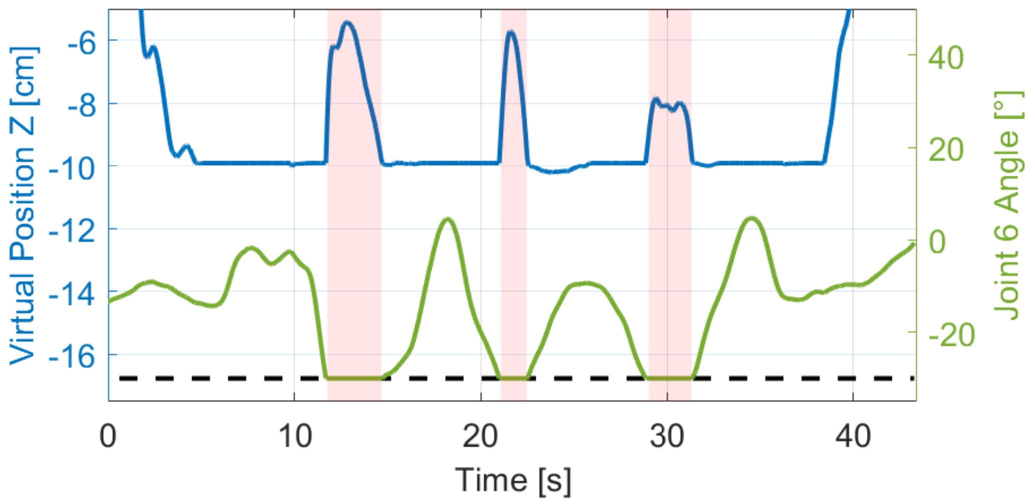


Figure 7.8: The instances (pale red) during which the virtual Z-position (blue) deviates from the constraint set (around -10cm) in the bottom left subfigure of Fig. 7.7 is caused by joint 6 (wrist flexion/extension) of the right arm (green) on the EXO-UL8 reaching its lower mechanical limit (dashed black line). Since the joint cannot physically go lower, the virtual position cannot maintain its set point. A similar case is evident in the α_2-orientation component of the virtual state.

The virtual constraints controller enables bimanual interaction with virtual rigid objects by constraining the relative pose between the operator's hands. The approach is agnostic to the geometry of the virtual object when constraining the motion, which has the benefit of being generalizable and places no restrictions on the shape of the object, such as convexity. The only place in which the object's geometry would be used is in detecting initial contact with it in order to record (p_0, α_0). However, detecting contact is much simpler than constraining dynamics to respect their geometries, which is typically done in force-based approaches. The proposed method also does not suffer from instability of contacting high stiffness objects, making it appropriate for virtual interactions in VR-based rehabilitation applications.

The use of second-order virtual dynamics in the admittance control attempts to simplify pHRI by leveraging our intuition of how rigid bodies move. In the virtual constraints controller, this choice manifests as the transversal output having vector relative degree of 2 due to the virtual constraints being holonomic. However, the method can extend to nonholonomic constraints with virtual dynamics of any order by following the same approach, as long as the virtual outputs remain input-output feedback linearizable.

7.7 Conclusion

This study introduces a novel feedback linearization-inspired control methodology for constraining admittance control to virtual motion subsets in exoskeleton-based rehabilitation applications. The study explores two specific applications: a path-guided reaching task commonly encountered in AAN rehabilitation, and bimanual interaction with a virtual object, often used in VR-based rehabilitation. In the former, the proposed method confines motion to an elliptical path and uses a virtual spring-damper restoring force field. Experiments explore the impact of adjusting the constraint strength parameter, which regulates the strictness of the motion constraint, serving as an analogue for the force field strength. In the latter experiment, the method rigidly constrains the end effectors of a bimanual exoskeleton in its 14-dimensional bimanual joint space in order to simulate interaction with a virtual object, and demonstrate a high-dimensional application for which local coordinates on the constraint manifold cannot easily be found. The two experiments illustrate the virtual constraint controller's robustness against operator-applied forces that can violate the constraint, while being transparent to forces that propagate motion within the constraint. The experiments exemplify markedly different applications within exoskeleton-based rehabilitation, showcasing the generalizability of the proposed virtually constrained admittance control methodology for constrained pHRI.

CHAPTER 8

Conclusion

This dissertation explored various challenges concerning pHRI with exoskeleton robots, including safety for HITL systems, admittance control for high transparency interactions, human input estimation, and constrained control for robot rehabilitation and VR-based applications. These challenges were explored through the following contributions:

1. **A reference level safety approach for pHRI with serial link manipulators:** Chapter 4 focused on the development of a safety approach that defines virtual soft and hard bounding regions for the admittance controller's virtual dynamics. A joint space collision avoidance algorithm is also presented, which ensures that all points along the manipulator are constrained from moving too closely to other bodies in order to avoid potential collisions. The entire methodology is also implemented and distributed as an open-source C++ library: https://github.com/jianwei-sun/gtfo.

2. **A study of rate-limiting as a potential mitigation to delay-induced instability:** Chapter 5 investigated the phenomenon of delay-induced instability, in which a HITL system can become unstable in the presence of time-delay, even though both the human operator and exoskeleton are stable systems by themselves. The study analyzed a rate-limiting filter placed after the human-applied forces and provided theoretical bounds on how the filter's rate-limiting threshold should be set. Experimental results verified the filter's performance in suppressing instability and allowing recovery.

3. **A method for reducing the number of force sensors while maintaining transparency:** Chapter 6 explored a method for utilizing the Kalman filter-based sensor fusion method using a strict subset of sensors, while maintaining comparable transparency. The methodology is verified with force sensing on the EXO-UL8, showing that one of the three sensors per arm can be removed, improving wearability of the exoskeleton.

4. **A feedback linearization-inspired method for enabling holonomically constrained admittance control:** Chapter 7 proposed a time-invariant admittance control methodology for enforcing virtual constraints that are modeled as level sets of smooth functions. The strength of the enforcement can be tuned via a single parameter. The utility of the approach and tuning is experimentally demonstrated on two important pHRI applications: path-following within AAN rehabilitation using the V-Rex, and bimanual interaction with virtual objects in VR using the EXO-UL8.

The field of pHRI is a large and multifaceted domain that gives rise to various challenges across robotics, control, and human factors. As robotics become more widespread and commonplace in society, it will undoubtedly overlap with many existing fields that are traditionally human-dominated, such as in robotic rehabilitation, robot surgery, warehouse automation, logistics, and exploration, to name a few. This growth has created many new challenges at the intersection of humans and robots. This dissertation explored and addressed some of these challenges in the aspects of safety, estimation, and control. However, this work is only one step toward realizing the maximum potential of human-robot interaction.

References

[1] J. Sun, Y. Foroutani, and J. Rosen, "Virtually constrained admittance control using feedback linearization for physical human-robot interaction with rehabilitation exoskeletons," *Under review.*

[2] J. Sun, E. H. Kramer, and J. Rosen, "A safety-focused admittance control approach for physical human-robot interaction with rigid multi-arm serial link exoskeletons," *Under review.*

[3] P. W. Ferguson, J. Sun, and J. Rosen, "On the other hand: A bimanual, reconfigurable hand exoskeleton with opposable thumbs for use with upper limb exoskeletons," *Under review.*

[4] J. Sun, P. W. Ferguson, and J. Rosen, "Suppressing delay-induced oscillations in physical human-robot interaction with an upper-limb exoskeleton using rate-limiting," in *2022 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 6695–6701, 2022.

[5] J. Sun, Y. Shen, and J. Rosen, "Sensor Reduction, Estimation, and Control of an Upper-Limb Exoskeleton," *IEEE Robotics and Automation Letters*, vol. 6, pp. 1012–1019, Apr. 2021.

[6] Y. Shen, J. Sun, J. Ma, and J. Rosen, "Admittance Control Scheme Comparison of EXO-UL8: A Dual-Arm Exoskeleton Robotic System," in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, pp. 611–617, June 2019.

[7] J. Sun and R. Gill, "Vehicle Platoon Control with Virtual Path Constraints," in *2019 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 456–461, Aug. 2019.

[8] C. Geaney, J. Sun, S. V. Hum, E. S. Rogers, E. Martinez-de Rioja, and J. A. Encinar, "Synthesis of a multi-beam dual reflectarray antenna using genetic algorithms," in *2017 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting*, pp. 1179–1180, 2017.

[9] C. Bower, H. Taheri, and E. Wolbrecht, "Adaptive control with state-dependent modeling of patient impairment for robotic movement therapy," in *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, pp. 1–6, 2013.

[10] N. Naghavi, A. Akbarzadeh, S. M. Tahamipour-Z., and I. Kardan, "Assist-As-Needed control of a hip exoskeleton based on a novel strength index," *Robotics and Autonomous Systems*, vol. 134, p. 103667, Dec. 2020.

[11] K. Gui, U.-X. Tan, H. Liu, and D. Zhang, "Electromyography-driven progressive assist-as-needed control for lower limb exoskeleton," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 1, pp. 50–58, 2020.

[12] E. T. Wolbrecht, V. Chan, D. J. Reinkensmeyer, and J. E. Bobrow, "Optimizing compliant, model-based robotic assistance to promote neurorehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 3, pp. 286–297, 2008.

[13] P. Agarwal and A. D. Deshpande, "Subject-specific assist-as-needed controllers for a hand exoskeleton for rehabilitation," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 508–515, 2018.

[14] H. J. Asl, T. Narikiyo, and M. Kawanishi, "An assist-as-needed control scheme for robot-assisted rehabilitation," in *2017 American Control Conference (ACC)*, pp. 198–203, 2017.

[15] M. Maaref, A. Rezazadeh, K. Shamaei, R. Ocampo, and T. Mahdi, "A bicycle cranking model for assist-as-needed robotic rehabilitation therapy using learning from demonstration," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 653–660, 2016.

[16] S. Pareek and T. Kesavadas, "iart: Learning from demonstration for assisted robotic therapy using lstm," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 477–484, 2020.

[17] A. Hace and M. Franc, "Pseudo-sensorless high-performance bilateral teleoperation by sliding-mode control and fpga," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 1, pp. 384–393, 2014.

[18] S. Akutsu, H. Sekiguchi, T. Nozaki, and T. Murakami, "Position and torque sensorless motion transmission for haptic teleoperation using two types of voltage compensation," in *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, pp. 1–6, 2017.

[19] R. S. Mosher, "Handyman to hardiman," *SAE Transactions*, vol. 76, pp. 588–597, 1968.

[20] S. Robotics, "Sarcos Achieves Unprecedented Power Performance Overcoming Major Obstacle to Commercial Deployment of Full-Body, Powered Industrial Exoskeletons," Nov. 2018.

[21] S. Marcheschi, F. Salsedo, M. Fontana, and M. Bergamasco, "Body extender: Whole body exoskeleton for human power augmentation," in *2011 IEEE International Conference on Robotics and Automation*, pp. 611–616, 2011.

[22] H. Arai, T. Takubo, Y. Hayashibara, and K. Tanie, "Human-robot cooperative manipulation using a virtual nonholonomic constraint," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 4, pp. 4063–4069 vol.4, 2000.

[23] J. DelPreto and D. Rus, “Sharing the load: Human-robot team lifting using muscle activity,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7906–7912, 2019.

[24] L. Rozo, S. Calinon, D. G. Caldwell, P. Jiménez, and C. Torras, “Learning physical collaborative robot behaviors from human demonstrations,” *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 513–527, 2016.

[25] L. Lu and J. T. Wen, “Human-robot cooperative control for mobility impaired individuals,” in *2015 American Control Conference (ACC)*, pp. 447–452, 2015.

[26] A. Okamura, “Methods for haptic feedback in teleoperated robot-assisted surgery,” *Industrial Robot: An International Journal*, vol. 31, pp. 499–508, Jan. 2004.

[27] J. Kofman, X. Wu, T. Luu, and S. Verma, “Teleoperation of a robot manipulator using a vision-based human-robot interface,” *IEEE Transactions on Industrial Electronics*, vol. 52, no. 5, pp. 1206–1219, 2005.

[28] F. Just, Ö. Özen, P. Bösch, H. Bobrovsky, V. Klamroth-Marganska, R. Riener, and G. Rauter, “Exoskeleton transparency: Feed-forward compensation vs. disturbance observer,” *at - Automatisierungstechnik*, vol. 66, pp. 1014–1026, Nov. 2018.

[29] H. Lee, B. Lee, W. Kim, M. Gil, J. Han, and C. Han, “Human-robot cooperative control based on pHRI (Physical Human-Robot Interaction) of exoskeleton robot for a human upper extremity,” *International Journal of Precision Engineering and Manufacturing*, vol. 13, pp. 985–992, June 2012.

[30] S. Tarbouriech, B. Navarro, P. Fraisse, A. Crosnier, A. Cherubini, and D. Sallé, “Admittance control for collaborative dual-arm manipulation,” in *2019 19th International Conference on Advanced Robotics (ICAR)*, pp. 198–204, 2019.

[31] M. Parigi Polverini, A. M. Zanchettin, and P. Rocco, “Real-time collision avoidance in human-robot interaction based on kinetostatic safety field,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4136–4141, 2014.

[32] E. Kaplanoglu, E. Erdemir, A. Jones, R. Bailey, K. Kawamura, and S. K. Hargrove, “Intelligent grasping classification method for a smart prosthetic hand,” in *SoutheastCon 2018*, pp. 1–5, 2018.

[33] M. Peshkin, J. Colgate, W. Wannasuphoprasit, C. Moore, R. Gillespie, and P. Akella, “Cobot architecture,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 377–390, 2001.

[34] “Autopilot and full self-driving capability,” Feb. 2019.

[35] “Moley robotics: The world’s first robotic kitchen,” Apr. 2019.

[36] “Kiva systems: Three engineers, hundreds of robots, one warehouse,” Apr. 2021.

[37] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing, London: Springer-Verlag, 2009.

[38] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Upper Saddle River, N.J: Pearson, 3rd edition ed., Aug. 2004.

[39] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ: Wiley, 1st edition ed., Nov. 2005.

[40] R. M. Murray, Z. Li, and S. Sastry, *A mathematical introduction to robotic manipulation*. Boca Raton: CRC Press, 1994.

[41] E. Cavallaro, J. Rosen, J. Perry, and S. Burns, "Real-Time Myoprocessors for a Neural Controlled Powered Exoskeleton Arm," *IEEE Transactions on Biomedical Engineering*, vol. 53, pp. 2387–2396, Nov. 2006.

[42] Z. Li, B. Wang, F. Sun, C. Yang, Q. Xie, and W. Zhang, "semg-based joint force control for an upper-limb power-assist exoskeleton robot," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 3, pp. 1043–1050, 2014.

[43] L. M. Miller, *Comprehensive Control Strategies for a Seven Degree of Freedom Upper Limb Exoskeleton Targeting Stroke Rehabilitation*. Thesis, University of Washington, Feb. 2013.

[44] C. Fleischer, A. Wege, K. Kondak, and G. Hommel, "Application of EMG signals for controlling exoskeleton robots," *Biomedizinische Technik/Biomedical Engineering*, vol. 51, pp. 314–319, Dec. 2006.

[45] D. Jayanth, E. S. Sai, and Y. Sangeetha, "Surface emg based intelligent signal processing for exoskeleton," in *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 76–82, 2017.

[46] M. Witkowski, M. Cortese, M. Cempini, J. Mellinger, N. Vitiello, and S. R. Soekadar, "Enhancing brain-machine interface (BMI) control of a hand exoskeleton using electrooculography (EOG)," *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 1, p. 165, 2014.

[47] T. Noda, N. Sugimoto, J. Furukawa, M.-a. Sato, S.-H. Hyon, and J. Morimoto, "Brain-controlled exoskeleton robot for bmi rehabilitation," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pp. 21–27, 2012.

[48] T. D. Lalitharatne, K. Teramoto, Y. Hayashi, and K. Kiguchi, "Evaluation of perception-assist with an upper-limb power-assist exoskeleton using emg and eeg signals," in *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control*, pp. 524–529, 2014.

[49] C.-C. Lien, "A scalable model-based hand posture analysis system," *Machine Vision and Applications*, vol. 16, pp. 157–169, May 2005.

[50] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, pp. 52–73, Oct. 2007.

[51] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Hand Pose Estimation and Hand Shape Classification Using Multi-layered Randomized Decision Forests," in *Computer Vision – ECCV 2012* (A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 852–863, Springer, 2012.

[52] S. J. Ball, I. E. Brown, and S. H. Scott, "Medarm: a rehabilitation robot with 5dof at the shoulder complex," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, pp. 1–6, 2007.

[53] R. Richardson, M. Brown, B. Bhakta, and M. Levesley, "Design and control of a three degree of freedom pneumatic physiotherapy robot," *Robotica*, vol. 21, pp. 589–604, Dec. 2003.

[54] L. Xia, Y. Feng, L. Zheng, C. Wang, and X. Wu, "Development of An Adaptive Iterative Learning Controller With Sensorless Force Estimator for The Hip-type Exoskeleton," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, (Dali, China), pp. 2516–2521, IEEE, Dec. 2019.

[55] B. Ugurlu, M. Nishimura, K. Hyodo, M. Kawanishi, and T. Narikiyo, "A framework for sensorless torque estimation and control in wearable exoskeletons," in *2012 12th IEEE International Workshop on Advanced Motion Control (AMC)*, (Sarajevo, Bosnia and Herzegovina), pp. 1–7, IEEE, Mar. 2012.

[56] Y. Kanai and Y. Fujimoto, "Performance Analysis of Torque-sensorless Assist Control of a Powered Exoskeleton Using Highly Back-drivable Actuators," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, (Helsinki, Finland), pp. 577–582, IEEE, July 2019.

[57] J. C. Perry, J. Rosen, and S. Burns, "Upper-limb powered exoskeleton design," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 4, pp. 408–417, 2007.

[58] Y. Shen, J. Ma, B. Dobkin, and J. Rosen, "Asymmetric Dual Arm Approach For Post Stroke Recovery Of Motor Functions Utilizing The EXO-UL8 Exoskeleton System: A Pilot Study," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, vol. 2018, pp. 1701–1707, July 2018.

[59] M. Simkins, N. Byl, H. Kim, G. Abrams, and J. Rosen, "Upper limb bilateral symmetric training with robotic assistance and clinical outcomes for stroke," *International Journal of Intelligent Computing and Cybernetics*, vol. 9, pp. 83–104, Jan 2016.

[60] H. Kim and J. Rosen, "Predicting redundancy of a 7 dof upper limb exoskeleton toward improved transparency between human and robot," *Journal of Intelligent & Robotic Systems*, vol. 80, pp. 99–119, Dec 2015.

[61] W. Yu and J. Rosen, "Neural pid control of robot manipulators with application to an upper limb exoskeleton," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 673–684, 2013.

[62] Y. Shen, B. P.-Y. Hsiao, J. Ma, and J. Rosen, "Upper limb redundancy resolution under gravitational loading conditions: Arm postural stability index based on dynamic manipulability analysis," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, (Birmingham), pp. 332–338, IEEE, Nov. 2017.

[63] F. Xiao, Y. Gao, Y. Wang, Y. Zhu, and J. Zhao, "Design and evaluation of a 7-DOF cable-driven upper limb exoskeleton," *Journal of Mechanical Science and Technology*, vol. 32, pp. 855–864, Feb. 2018.

[64] D. Ma, S. Yan, Z. Yin, and Y. Yang, "Investigation of the friction behavior of harmonic drive gears at low speed operation," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1382–1388, 2018.

[65] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, v. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz, "Sympy: symbolic computing in python," *PeerJ Computer Science*, vol. 3, p. e103, Jan. 2017.

[66] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

[67] F. Dimeas and N. Aspragathos, "Online stability in human-robot cooperation with admittance control," *IEEE Transactions on Haptics*, vol. 9, no. 2, pp. 267–278, 2016.

[68] F. Ferraguti, C. Talignani Landi, L. Sabattini, M. Bonfè, C. Fantuzzi, and C. Secchi, "A variable admittance control strategy for stable physical human–robot interaction," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 747–765, 2019.

[69] A. Sharkawy, P. Koustoumpardis, and N. Aspragathos, "A neural network-based approach for variable admittance control in human–robot cooperation: online adjustment of the virtual inertia," *Intelligent Service Robotics*, 2020.

[70] F. Augugliaro and R. D'Andrea, "Admittance control for physical human-quadrocopter interaction," in *2013 European Control Conference (ECC)*, (Zurich), pp. 1805–1810, IEEE, July 2013.

[71] D. Simon, *Optimal State Estimation: Kalman, H Infinity and Nonlinear Approaches*. Hoboken, N.J: Wiley-Interscience, 2006.

[72] J. L. Speyer, W. H. Chung, and Society for Industrial and Applied Mathematics, *Stochastic processes, estimation, and control*. Philadelphia, Pa.: Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2008.

[73] J. A. Martinez, P. Ng, S. Lu, M. S. Campagna, and O. Celik, "Design of wrist gimbal: A forearm and wrist exoskeleton for stroke rehabilitation," in *2013 IEEE 13th Int. Conf. on Rehabilitation Robotics (ICORR)*, pp. 1–6, 2013.

[74] S. Wang, L. Wang, C. Meijneke, E. van Asseldonk, T. Hoellinger, G. Cheron, Y. Ivanenko, V. La Scaleia, F. Sylos-Labini, M. Molinari, F. Tamburella, I. Pisotta, F. Thorsteinsson, M. Ilzkovitz, J. Gancet, Y. Nevatia, R. Hauffe, F. Zanow, and H. van der Kooij, "Design and control of the mindwalker exoskeleton," *IEEE Trans. on Neural Systems and Rehabilitation Eng.*, vol. 23, no. 2, pp. 277–286, 2015.

[75] Y. Long, Z. Du, C. feng Chen, W. Wang, L. He, X.-W. Mao, G. Xu, G. Zhao, and W. Dong, "Hybrid control scheme of a hydraulically actuated lower extremity exoskeleton for load-carrying," *Journal of Intelligent & Robotic Systems*, vol. 91, pp. 493–500, 2018.

[76] P. Beyl, K. Knaepen, S. Duerinck, M. Damme, B. Vanderborght, R. Meeusen, and D. Lefeber, "Safe and compliant guidance by a powered knee exoskeleton for robot-assisted rehabilitation of gait," *Advanced Robotics*, vol. 25, pp. 513–535, 2011.

[77] H. Liao, H. H.-T. Chan, F. Gao, X. Zhao, G. Liu, and W.-H. Liao, "Proxy-based torque control of motor-driven exoskeletons for safe and compliant human-exoskeleton interaction," *Mechatronics*, vol. 88, 2022.

[78] M. J. Kim, W. Lee, C. Ott, and W. K. Chung, "A passivity-based admittance control design using feedback interconnections," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 801–807, IEEE, 2016.

[79] D. Liu, J. Xu, C. Chen, X. Long, D. Tao, and X. Wu, "Vision-assisted autonomous lower-limb exoskeleton robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3759–3770, 2021.

[80] Y. Cho, K. Kim, S. Ma, and J. Ueda, "A robotic wearable exoskeleton for construction worker's safety and health," in *Construction Research Congress 2018*, pp. 19–28, 2018.

[81] H. Kim and W. Yang, "Variable Admittance Control Based on Human–Robot Collaboration Observer Using Frequency Analysis for Sensitive and Safe Interaction," *Sensors*, vol. 21, no. 5, p. 1899, 2021.

[82] A. Vick, D. Surdilovic, and J. Kruger, "Safe physical human-robot interaction with industrial dual-arm robots," in *9th International Workshop on Robot Motion and Control*, pp. 264–269, IEEE, 2013.

[83] H. Rifaï, S. Mohammed, W. Hassani, and Y. Amirat, "Nested saturation based control of an actuated knee joint orthosis," *Mechatronics*, vol. 23, no. 8, pp. 1141–1149, 2013.

[84] T. Zhang, M. Tran, and H. Huang, "Admittance Shaping-Based Assistive Control of SEA-Driven Robotic Hip Exoskeleton," *IEEE/ASME Trans. Mechatron.*, vol. 24, no. 4, pp. 1508–1519, 2019.

[85] M. Li, M. Ishii, and R. H. Taylor, "Spatial motion constraints using virtual fixtures generated by anatomy," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 4–19, 2007.

[86] D. Bazzi, F. Roveda, A. M. Zanchettin, and P. Rocco, "A unified approach for virtual fixtures and goal-driven variable admittance control in manual guidance applications," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6378–6385, 2021.

[87] L. Žlajpah and T. Petrič, "Unified virtual guides framework for path tracking tasks," *Robotica*, vol. 38, no. 10, p. 1807–1823, 2020.

[88] W. He, C. Xue, X. Yu, Z. Li, and C. Yang, "Admittance-Based Controller Design for Physical Human–Robot Interaction in the Constrained Task Space," *IEEE Transactions on Automation Science and Engineering*, vol. 17, pp. 1937–1949, Oct. 2020.

[89] R. Moccia, C. Iacono, B. Siciliano, and F. Ficuciello, "Vision-based dynamic virtual fixtures for tools collision avoidance in robotic surgery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1650–1655, 2020.

[90] A. De Luca and F. Flacco, "Integrated control for phri: Collision avoidance, detection, reaction and collaboration," in *2012 4th IEEE RAS & EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, pp. 288–295, 2012.

[91] Z. Li, H. Wei, G. Liu, C. Liu, and D. Chen, "A variable admittance control strategy for self-collision avoidance based on virtual constraints," in *2022 IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, pp. 958–963, 2022.

[92] P. Bosscher and D. Hedman, "Real-time collision avoidance algorithm for robotic manipulators," *Industrial Robot*, vol. 38, no. 2, pp. 186–197, 2011.

[93] A. Dietrich, T. Wimböck, H. Täubig, A. Albu-Schäffer, and G. Hirzinger, "Extensions to reactive self-collision avoidance for torque and position controlled humanoids," in *2011 IEEE Int. Conf. on Robotics and Automation*, pp. 3455–3462, 2011.

[94] H. Liu, D. Qu, F. Xu, Z. Du, J. Song, and M. Liu, "Real-time and efficient collision avoidance planning approach for safe human-robot interaction," *Journal of Intelligent & Robotic Systems*, vol. 105, 2022.

[95] H. Lin, Y. Fan, T. Tang, and M. Tomizuka, "Human guidance programming on a 6-dof robot with collision avoidance," in *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2676–2681, 2016.

[96] H. Kaneko, T. Arai, K. Inoue, and Y. Mae, "Real-time obstacle avoidance for robot arm using collision jacobian," in *1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 617–622 vol.2, 1999.

[97] Z. Zhang, L. Zheng, Z. Chen, L. Kong, and H. R. Karimi, "Mutual-collision-avoidance scheme synthesized by neural networks for dual redundant robot manipulators executing cooperative tasks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 3, pp. 1052–1066, 2021.

[98] P. Chembrammel and T. Kesavadas, "A new implementation for online calculation of manipulator jacobian," *PLOS ONE*, vol. 14, 2019.

[99] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *Robotics Research* (S. Thrun, R. Brooks, and H. Durrant-Whyte, eds.), (Berlin, Heidelberg), pp. 49–64, Springer Berlin Heidelberg, 2007.

[100] D. Zhang, G. Yang, and R. P. Khurshid, "Haptic teleoperation of uavs through control barrier functions," *IEEE Transactions on Haptics*, vol. 13, no. 1, pp. 109–115, 2020.

[101] Y. He, Y. Hu, P. Zhang, B. Zhao, X. Qi, and J. Zhang, "Human–Robot Cooperative Control Based on Virtual Fixture in Robot-Assisted Endoscopic Sinus Surgery," *Applied Sciences*, vol. 9, p. 1659, Apr. 2019.

[102] Y. Shen and J. Rosen, "Chapter 5 - exo-ul upper limb robotic exoskeleton system series: From 1 dof single-arm to (7+1) dofs dual-arm," in *Wearable Robotics* (J. Rosen and P. W. Ferguson, eds.), pp. 91–103, Academic Press, 2020.

[103] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "Osqp: An operator splitting solver for quadratic programs," pp. 339–339, 2018.

[104] G. Guennebaud, B. Jacob, *et al.*, "Eigen v3." http://eigen.tuxfamily.org, 2010.

[105] A. Q. Keemink, H. van der Kooij, and A. H. Stienen, "Admittance control for physical human–robot interaction," *The International Journal of Robotics Research*, vol. 37, pp. 1421–1444, Sept. 2018.

[106] S. F. Atashzar, H.-Y. Huang, F. D. Duca, E. Burdet, and D. Farina, "Energetic passivity decoding of human hip joint for physical human-robot interaction," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5953–5960, 2020.

[107] D. Buongiorno, D. Chiaradia, S. Marcheschi, M. Solazzi, and A. Frisoli, "Multi-DoFs Exoskeleton-Based Bilateral Teleoperation with the Time-Domain Passivity Approach," *Robotica*, vol. 37, no. 9, pp. 1641–1662, 2019.

[108] U. Keller, H. J. A. van Hedel, V. Klamroth-Marganska, and R. Riener, "ChARMin: The First Actuated Exoskeleton Robot for Pediatric Arm Rehabilitation," *IEEE/ASME Trans. Mechatron.*, vol. 21, no. 5, pp. 2201–2213, 2016.

[109] A. M. Khan, D.-w. Yun, M. A. Ali, K. M. Zuhaib, C. Yuan, J. Iqbal, J. Han, K. Shin, and C. Han, "Passivity based adaptive control for upper extremity assist exoskeleton," *Int. J. Control Autom. Syst.*, vol. 14, no. 1, pp. 291–300, 2016.

[110] D. L. Castro, C.-H. Zhong, F. Braghin, and W.-H. Liao, "Lower Limb Exoskeleton Control via Linear Quadratic Regulator and Disturbance Observer," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1743–1748, IEEE, 2018.

[111] C. Chen, S. Zhang, X. Zhu, J. Shen, and Z. Xu, "Disturbance Observer-Based Patient-Cooperative Control of a Lower Extremity Rehabilitation Exoskeleton," *Int. J. Precis. Eng. Manuf.*, vol. 21, no. 5, pp. 957–968, 2020.

[112] H.-B. Kang and J.-H. Wang, "Adaptive control of 5 dof upper-limb exoskeleton robot with improved safety," *ISA Transactions*, vol. 52, no. 6, pp. 844–852, 2013.

[113] O. Baser, H. Kizilhan, and E. Kilic, "Employing variable impedance (stiffness/damping) hybrid actuators on lower limb exoskeleton robots for stable and safe walking trajectory tracking," *Journal of Mechanical Science and Technology*, vol. 34, no. 6, pp. 2597–2607, 2020.

[114] D. H. Klyde, D. T. McRuer, and T. T. Myers, "Pilot-Induced Oscillation Analysis and Prediction with Actuator Rate Limiting," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 81–89, 1997.

[115] T. K. Mandal and Y. Gu, "Analysis of Pilot-Induced-Oscillation and Pilot Vehicle System Stability Using UAS Flight Experiments," *Aerospace*, vol. 3, p. 42, Dec. 2016.

[116] S. L. Gatley, M. C. Turner, I. Postlethwaite, and A. Kumar, "A comparison of rate-limit compensation schemes for pilot-induced-oscillation avoidance," *Aerospace Science and Technology*, vol. 10, no. 1, pp. 37–47, 2006.

[117] J. Yuan, S. Fei, and Y. Chen, "Compensation strategies based on Bode step concept for actuator rate limit effect on first-order plus time-delay systems," *Nonlinear Dynamics*, vol. 99, no. 4, pp. 2851–2866, 2020.

[118] X.-h. Liang, K. Yamada, N. Sakamoto, and I. Jikuya, "Model Predictive Controller Design to Suppress Rate-Limiter-Based Pilot-Induced Oscillations," *Trans. Japan Soc. Aero. S Sci.*, vol. 49, no. 166, pp. 239–245, 2007.

[119] P. F. Vint, S. P. Mclean, and G. M. Harron, "Electromechanical delay in isometric actions initiated from nonresting levels," *Medicine and Science in Sports and Exercise*, vol. 33, no. 6, pp. 978–983, 2001.

[120] T. Abuhamdia and J. Rosen, "Constant Visual and Haptic Time Delays in Simulated Bilateral Teleoperation: Quantifying the Human Operator Performance," *Presence: Teleoperators and Virtual Environments*, vol. 22, no. 4, pp. 271–290, 2013.

[121] R. Shadmehr and F. A. Mussa-Ivaldi, "Adaptive representation of dynamics during learning of a motor task," *Journal of Neuroscience*, vol. 14, pp. 3208–3224, May 1994.

[122] C. Nguiadem, M. Raison, and S. Achiche, "Motion Planning of Upper-Limb Exoskeleton Robots: A Review," *Applied Sciences*, vol. 10, no. 21, p. 7626, 2020.

[123] L. I. Lugo-Villeda, A. Frisoli, O. Sandoval-Gonzalez, M. A. Padilla, V. Parra-Vega, C. A. Avizzano, E. Ruffaldi, and M. Bergamasco, "Haptic guidance of Light-Exoskeleton for arm-rehabilitation tasks," in *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 903–908, IEEE, 2009.

[124] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of Neuroscience*, vol. 5, pp. 1688–1703, July 1985.

[125] S. Nagai, R. Oboe, T. Shimono, and A. Kawamura, "Fast Force Control without Force Sensor Using Combination of aaKF and RFOB for In-circuit Test with Probing System," *IEEJ Journal of Industry Applications*, vol. 8, pp. 152–159, Mar. 2019.

[126] A. Zare, H. Mohammadi, N. K. Dhingra, T. T. Georgiou, and M. R. Jovanovic, "Proximal Algorithms for Large-Scale Statistical Modeling and Sensor/Actuator Selection," *IEEE Transactions on Automatic Control*, vol. 65, pp. 3441–3456, Aug. 2020.

[127] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor selection for Kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms," *Automatica*, vol. 78, pp. 202–210, Apr. 2017.

[128] S. Joshi and S. Boyd, "Sensor Selection via Convex Optimization," *IEEE Transactions on Signal Processing*, vol. 57, pp. 451–462, Feb. 2009.

[129] T.-H. S. Li, Y.-T. Su, S.-H. Liu, J.-J. Hu, and C.-C. Chen, "Dynamic balance control for biped robot walking using sensor fusion, kalman filter, and fuzzy logic," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4394–4408, 2012.

[130] D. Novak and R. Riener, "A survey of sensor fusion methods in wearable robotics," *Robotics and Autonomous Systems*, vol. 73, pp. 155–170, Nov. 2015.

[131] C. Giraud and B. Jouvencel, "Sensor selection in a fusion process: a fuzzy approach," in *Proceedings of 1994 IEEE International Conference on MFI '94. Multisensor Fusion and Integration for Intelligent Systems*, (Las Vegas, NV, USA), pp. 599–606, IEEE, 1994.

[132] M. Vemula and P. M. Djuric, "Multisensor fusion for target tracking using sequential monte carlo methods," in *IEEE/SP 13th Workshop on Statistical Signal Processing, 2005*, pp. 1304–1309, July 2005.

[133] M. Kawamura, S. Ookubo, Y. Asano, T. Kozuki, K. Okada, and M. Inaba, "A joint-space controller based on redundant muscle tension for multiple DOF joints in musculoskeletal humanoids," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, (Cancun, Mexico), pp. 814–819, IEEE, Nov. 2016.

[134] Z. Xu, S. X. Yang, and S. A. Gadsden, "Enhanced Bioinspired Backstepping Control for a Mobile Robot With Unscented Kalman Filter," *IEEE Access*, vol. 8, pp. 125899–125908, 2020.

[135] N. Jarrasse and G. Morel, "Connecting a Human Limb to an Exoskeleton," *IEEE Transactions on Robotics*, vol. 28, pp. 697–709, June 2012.

[136] Hyunchul Kim, L. M. Miller, N. Byl, G. Abrams, and J. Rosen, "Redundancy Resolution of the Human Arm and an Upper Limb Exoskeleton," *IEEE Transactions on Biomedical Engineering*, vol. 59, pp. 1770–1779, June 2012.

[137] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," in *Advances in Psychology* (P. A. Hancock and N. Meshkati, eds.), vol. 52 of *Human Mental Workload*, pp. 139–183, North-Holland, Jan. 1988.

[138] P. Lee, S. Wei, J. Zhao, and N. I. Badler, "Strength guided motion," in *Proceedings of the 17th annual conference on Computer graphics and interactive techniques - SIGGRAPH '90*, (Dallas, TX, USA), pp. 253–262, ACM Press, 1990.

[139] Y. Shen, P. W. Ferguson, and J. Rosen, "Chapter 1 - upper limb exoskeleton systems—overview," in *Wearable Robotics* (J. Rosen and P. W. Ferguson, eds.), pp. 1–22, Academic Press, 2020.

[140] O. Mubin, F. S. Alnajjar, N. Jishtu, B. S. Alsinglawi, and A. A. Mahmud, "Exoskeletons with virtual reality, augmented reality, and gamification for stroke patients' rehabilitation: Systematic review," *JMIR Rehabilitation and Assistive Technologies*, vol. 6, 2019.

[141] W. Qi, S. Sun, T. Niu, and D. Zhao, "Research and prospects of virtual reality systems applying exoskeleton technology," *Universal Access in the Information Society*, pp. 1–22, 10 2022.

[142] Q. Ai, Z. Liu, W. Meng, Q. Liu, and S. Q. Xie, "Machine learning in robot assisted upper limb rehabilitation: A focused review," *IEEE Transactions on Cognitive and Developmental Systems*, pp. 1–1, 2021.

[143] U. P. Matic Trlep, Matjaž Mihelj and M. Munih, "Rehabilitation robot with patient-cooperative control for bimanual training of hemiparetic subjects," *Advanced Robotics*, vol. 25, no. 15, pp. 1949–1968, 2011.

[144] E. D'Antonio, E. Galofaro, F. Patané, M. Casadio, and L. Masia, "A dual arm haptic exoskeleton for dynamical coupled manipulation," in *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1237–1242, 2021.

[145] M. Csikszentmihalyi, *Flow and the Foundations of Positive Psychology: The Collected Works of Mihaly Csikszentmihalyi.* 04 2014.

[146] H. J. Asl, M. Yamashita, T. Narikiyo, and M. Kawanishi, "Field-based assist-as-needed control schemes for rehabilitation robots," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 4, pp. 2100–2111, 2020.

[147] L. Pezeshki, H. Sadeghian, M. Keshmiri, X. Chen, and S. Haddadin, "Cooperative assist-as-needed control for robotic rehabilitation: A two-player game approach," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2852–2859, 2023.

[148] A. Duschau-Wicke, J. Von Zitzewitz, A. Caprez, L. Lunenburger, and R. Riener, "Path control: a method for patient-cooperative robot-aided gait rehabilitation," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 18, no. 1, pp. 38–48, 2009.

[149] D. Shi, W. Zhang, W. Zhang, and X. Ding, "Assist-as-needed attitude control in three-dimensional space for robotic rehabilitation," *Mechanism and Machine Theory*, vol. 154, p. 104044, 2020.

[150] X. Jin, X. Cui, and S. K. Agrawal, "Design of a cable-driven active leg exoskeleton (c-alex) and gait training experiments with human subjects," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5578–5583, IEEE, 2015.

[151] H. J. Asl, T. Narikiyo, and M. Kawanishi, "An assist-as-needed velocity field control scheme for rehabilitation robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3322–3327, IEEE, 2018.

[152] B. W. Gasser, A. Martínez, E. Sasso-Lance, C. Kandilakis, C. M. Durrough, and M. Goldfarb, "Preliminary assessment of a hand and arm exoskeleton for enabling bimanual tasks for individuals with hemiparesis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 10, pp. 2214–2223, 2020.

[153] B. Kim and A. D. Deshpande, "An upper-body rehabilitation exoskeleton harmony with an anatomical shoulder mechanism: Design, modeling, control, and performance evaluation," *The International Journal of Robotics Research*, vol. 36, no. 4, pp. 414–435, 2017.

[154] A. Frisoli, A. Montagner, L. Borelli, F. Salsedo, and M. Bergamasco, "A force-feedback exoskeleton for upper-limb rehabilitation in virtual reality," *Applied Bionics and Biomechanics*, vol. 6, pp. 115–126, 07 2009.

[155] A. Steed, S. Friston, V. Pawar, and D. Swapp, "Docking haptics: Extending the reach of haptics by dynamic combinations of grounded and worn devices," in *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology*, VRST '20, (New York, NY, USA), Association for Computing Machinery, 2020.

[156] J. Bender, M. Müller, M. Otaduy, and M. Teschner, "Position-based methods for the simulation of solid objects in computer graphics," 01 2013.

[157] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1321–1326, 2013.

[158] M. Selvaggio, G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, "Passive virtual fixtures adaptation in minimally invasive robotic surgery," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3129–3136, 2018.

[159] M. M. Marinho, H. Ishida, K. Harada, K. Deie, and M. Mitsuishi, "Virtual fixture assistance for suturing in robot-aided pediatric endoscopic surgery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 524–531, 2020.

[160] J. Abbott and A. Okamura, "Pseudo-admittance bilateral telemanipulation with guidance virtual fixtures," in *2006 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 169–175, 2006.

[161] S. Kana, K.-P. Tee, and D. Campolo, "Human–Robot co-manipulation during surface tooling: A general framework based on impedance control, haptic rendering and discrete geometry," *Robotics and Computer-Integrated Manufacturing*, vol. 67, p. 102033, Feb. 2021.

[162] R. S. D'Souza and C. Nielsen, "An algorithm for local transverse feedback linearization," *SIAM Journal on Control and Optimization*, vol. 61, no. 3, pp. 1248–1272, 2023.

[163] R. J. Gill, D. Kulić, and C. Nielsen, "Spline path following for redundant mechanical systems," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1378–1392, 2015.

[164] B. Maric, A. Mutka, and M. Orsag, "Collaborative human-robot framework for delicate sanding of complex shape surfaces," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2848–2855, 2020.

[165] J. E. Solanes, L. Gracia, P. Muñoz-Benavent, A. Esparza, J. Valls Miro, and J. Tornero, "Adaptive robust control and admittance control for contact-driven robotic surface conditioning," *Robotics and Computer-Integrated Manufacturing*, vol. 54, pp. 115–132, 2018.

[166] A. Isidori, M. Thoma, E. D. Sontag, B. W. Dickinson, A. Fettweis, J. L. Massey, and J. W. Modestino, *Nonlinear Control Systems*. Berlin, Heidelberg: Springer-Verlag, 3rd ed., 1995.

[167] H. Martiros, A. Miller, N. Bucki, B. Solliday, R. Kennedy, J. Zhu, T. Dang, D. Pattison, H. Zheng, T. Tomic, P. Henry, G. Cross, J. VanderMey, A. Sun, S. Wang, and K. Holtz, "SymForce: Symbolic Computation and Code Generation for Robotics," in *Proceedings of Robotics: Science and Systems*, 2022.