Technische Universität München
TUM School of Computation, Information and Technology

Variable Impedance Control for Compliant Robot Interactions: An Approach Based on Energy Tanks, Learning from Demonstration, and Dynamical Systems

Youssef Michel Sadek Abdelwadoud, M.Sc.

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften (Dr.-Ing.) genehmigten Dissertation.

Vorsitz: Prof. Dr. Wolfgang Utschick

Prüfende der Dissertation:

1. Prof. Dr. Gordon Cheng
2. Prof. Dr. Etienne Burdet
3. Prof. Dr. Aleš Ude

Die Dissertation wurde am 13.11.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 10.06.2024 angenommen.

To my parents

"You wouldn't say anything to me, nor me to you. But we'd both know that you'd made it, that you were happy." -The Dark Knight Rises, 2012.

Acknowledgments

It has been five years since I started my PhD, and fair to say, it has been a unique experience, one that I will always look back at and cherish, perhaps with a small smile :). A PhD is a long and grueling journey, a roller coaster where one goes through a whole of spectrum of emotions. Luckily, I have been blessed with many great people along the way. I would like to start by thanking Prof. Dongheui Lee, for providing me the opportunity to start my thesis at the chair of Human-Centered Assistive Robotics (HCR), for the freedom she gave me to pursue my research interests, and for all her guidance and support. I am grateful to Prof. Gordon Cheng, for all his patience, trust, and support especially during the final stages of this work, as well as for creating a vibrant and pleasant working atmosphere at the Institute of Cognitive Systems (ICS). Special thanks also to Prof. Etienne Burdet, whom I am really honored to have as an examiner for my thesis. Prof. Etienne's seminal works on variable impedance control has been a true source of inspiration for my thesis. My appreciation goes to Prof. Aleš Ude for being also my examiner, and for his contribution to Learning-from-Demonstration, which deeply enriched my understanding of the topic. I would like to thank Prof. Wolfgang Utschick for being the chairperson for my thesis, and for handling all the administrative work associated with the thesis submission process.

Next, I would like to express my gratitude to all my collaborators throughout the past 5 years, without whom may of the works presented in this thesis would have not been possible. First, I would like to thank Prof. Christian Ott, who was someone I always looked up to, and admired his vast knowledge in control theory. I had the pleasure of coauthoring two papers with Christian, including a TRO. My deepest gratitude goes to Dr. Matteo Saveriano, first for helping me getting settled during my first months at HCR, for our fruitful collaborations that resulted in three papers, and the countless engaging discussions. Matteo also introduced me to Dr. Fares Abu-Dakka, who I enjoyed a lot working and collaborating with. Special thanks to Dr. Harsimran Singh and Dr. Julian Klodmann from the German Aerospace Center (DLR) for our collaboration on surgical robotics and for all the stimulating discussions we had. I am happy to have met Dr. Rahaf Rahal, who has been someone I really enjoyed collaborating with, and more importantly, a good friend. Through Rahaf, I had the pleasure of working with Dr. Claudio Pacchierotti and Prof. Paolo Robuffo Giordano. I would like to thank Dr. Leif Johannsen for our collaboration during the SPP project, the many interesting discussions, and for his valuable advice and guidance. Special thanks also to Dr. Nico Mansfeld, who was my supervisor during my Master thesis, but we still managed to stay in touch and he was never late to provide me with help and advice whenever I needed. I want to thank Maged

Iskandar for all the discussions we had whenever I visited DLR, which I always enjoyed.
Special thanks also to my former and current office mates and colleagues at HCR. I would like to start by thanking Katrin Schulleri, my companion throughout this journey. Katrin and I started our PhDs almost at the same time, and together we went through the whole PhD experience with all its ups and downs, and I am glad we finally made it. I also had the pleasure of working with Katrin, whether on the SPP project, the publications we collaborated on, or during teaching. I always enjoyed our conversations, work related or not. Special thanks also to Thomas Eiband, for all the interesting discussions and fun times we had on during many late evenings we spent together in the office, and for offering his help whenever I needed. I want to also express my gratitude to Karna Potwar, Christoph Willibald, Esteve Valls Mascaro, Camille Vindolet, Dan Huang, Dr. David Paulius, Dr. Shile Li and Matteo Pantano. Thank you also to Dr. Hyemin Ahn for the positive energy and all the fun discussions we had, and to Dr. Alejandro Agostini for being my mentor and for offering me with advice and guidance. I am grateful also to my ICS colleagues Simon Armleder, Nicolas Berberich, Dr. Florian Bergner, Dr. Julio Rogelio Guadarrama Olvera, Alireza Malekmohammadi, Dr. John Nassour, Natalia Paredes Acuna, Wenlan Shen, Constantin Uhde, Fengyi Wang and Annika Guez. Thank you to Katharina Stadler for her continuous help and support in the hardware maintenance in our lab. I am very thankful to Ilona Nar-Witte, Viviana Wittemeier, Brigitte Rosenlehner and Wibke Borngesser for their time, patience and for handling all the bureaucratic and administrative aspects in our chair. I truly appreciate the efforts of Julio Rogelio Guadarrama Olvera, Katrin Schulleri, Simon Armleder, Camille Vindolet, Annika Guez, Natalia Paredes Acuna, Nico Mansfeld and Matteo Saveriano in giving me valuable feedback on my thesis manuscript. Special thanks to my current and former students Xiao Chen, Haotian Xue, Riccardo Arduini and Youssef Abdelhalem for their contributions and entrusting me to supervise their Master thesis.
Outside the work environment, I would like to thank my friends for all the great times and memories we shared together. Thank you to Royia, Farah, Radwa, Omar, Mariam, (second) Omar, Leena, Hazem, Hisham, Nader, Sherouk, Nada, Sultana, Sophie, Caroline, Gabriel and Anastasia for all the hikes, game nights, dinners, dish parties, camping trips and fun (often deep ;)) deep conversations. Special thanks also my old friends back home Lobna, Ibrahim, Huissen, Mina, and Abdelrahman for their continuous support and for making my trips to Egypt fun and memorable.
Last, but certainly not least, I would like to thank my family for being an infinite source of love and emotional support. I want to thank my sister, Mary, for being one of the kindest persons ever, for always checking on me and for all the great memories and long conversations we had during her visits to Munich. I want to thank my brother Daniel, for his love and for always being there, for our long and intresting conversations especially during our therapeutic late night car drives, and for sharing my love to Pink Floyd. I would like to thank my father, for his neverending love and support not only during this thesis, but throughout my life. Without you, I wouldnt be here today. Thank you for being the sound of reason that I could always turn to whenever I needed advice or support, especially during very delicate stages of my thesis. To my mother: my ultimate source of support, love, care and kindness, thank you for always being there, thank you for doing everything you can to see me become the best version of myself and thank you, for always believing in me. It all goes back to you.
Above all, I would like to thank God, for endowing me with the strength, the passion and the perseverance to make it all the way to the end, and without which, this work would certainly not be possible.
I gratefully acknowledge the funding of the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) SPP priority program the Active Self.

Munich, 06.11.2023 Youssef Michel

Abstract

Modern-day robotics has witnessed a paradigm shift, transitioning from rigid, high-gain position-controlled robots used in structured industrial environments to more compliant and lightweight systems. This transition brings robots one step closer to everyday interaction and collaboration with humans. However, it also poses several challenges in designing algorithms that can enable robots to continuously adapt their behaviors to dynamic environments, interact safely and compliantly with their surroundings, and accomplish the desired tasks.

In this regard, the concept of Impedance Control is often used, where the idea is to modulate the robot dynamic behavior at the ports of interaction. This helps reducing the effect of position uncertainties and avoid large contact forces, thereby allowing for smooth and compliant interaction. Interestingly, humans adopt a similar strategy to control their interactive behavior. In addition, humans excel in their ability to interact proficiently with different environments, through the continuous modulation of their end-point force and impedance, achieved by the concurrent co-activation of suitable muscle pairs.

Inspired by that, this thesis primarily focuses on the development of variable impedance control and learning algorithms to ensure safe and adaptive physical interactions, for robots operating autonomously and in teleoperated settings. One of the main objectives in this thesis is concerned with guaranteeing the stable interaction of the robot with unknown passive environments, when driven by a variable impedance control law. In this regard, we develop a passive hierarchical variable impedance controller, in order to enable a redundant robot accomplish multiple tasks prioritized in a hierarchical manner, where each task is specified by time-varying impedance parameters. The stability treatment is approached within the framework of passivity theory, where stability is guaranteed by ensuring that no internal generation of energy happens in the system. To this end, we leverage the concept of energy tanks, where the idea is to allocate a specific energy budget that can be reserved for potentially non-passive control actions, thereby ensuring that the energy these actions can inject into the system remains bounded.

While stability is an important requirement for safety, it does not necessarily preclude unsafe robot motions. A major research focus in this thesis is the development of inherently safe motion generators. Unfortunately, the traditional approach of controlling the robot to follow a desired motion typically involves an open-loop configuration, where motion generation and impedance control are programmed as separate loops. This can pose serious safety issues, especially in uncertain and dynamic environments. To overcome this problem, a closed-loop configuration control strategy is developed, where motion generation and variable impedance control loops are combined into one integrated loop. This eliminates the problem of time-indexed trajectory tracking, making the robot safe and robust to perturbations or possible uncertainties. The developed controller, called Variable Stiffness Dynamical Systems (VSDS), is additionally utilized in shared control scenarios and extended to orientation tasks.
Another objective pursued in this thesis is developing methods to improve the performance of teleoperated contact tasks, while also reducing the cognitive and physical workloads typically associated with robot teleoperation. In this endeavor, first, we develop a variable impedance bilateral teleoperation architecture, where we leverage Learning from Demonstration to learn a stiffness adaptation policy from human demonstrations. The learned model shapes the remote robot's behavior during physical contact with the remote environment, for a compliant interaction. The developed approach serves as a basis for a shared control architecture envisioned for contact tasks that require simultaneous adaptation of the robot's motion, force and impedance. Task models learnt from humans are exploited to build virtual fixtures that can haptically guide the human towards the optimal task execution.
In summary, this work leverages variable impedance control and learning, to endow robots with safety and adaptability during physical interactions, in diverse scenarios, ranging from autonomous robot operation, to teleoperated shared control.

Contents

Abbreviations and Symbols

In this thesis, we write scalar quantities as plain letters, e.g., η, a, b. We use bold letters to represent vectors and matrices, e.g. $\boldsymbol{x}$, $\boldsymbol{K}(t)$. The derivative of a quantity with respect to time is represented by a dot on top of the symbol , i.e., $\dot{\boldsymbol{x}} = \frac{d}{dt}\boldsymbol{x}$, $\ddot{\boldsymbol{x}} = \frac{d^2}{dt^2}\boldsymbol{x}$. Typically, we use $\hat{\boldsymbol{x}}$ to denote the estimate of $\boldsymbol{x}$, and $\tilde{\boldsymbol{x}}$ to denote the error between $\boldsymbol{x}$ and a reference state. Finally, we have $||\boldsymbol{x}|| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$ indicate the L_2 norm, and $|\boldsymbol{x}|$ for the element-wise absolute values.
Next, a list of abbreviations and symbols is provided. Please note, the list is not exhaustive, but only introduces the abbreviations and symbols that are important or appear frequently in this thesis.

List of Symbols

i, j	Indices for numbering
t	Time
$\boldsymbol{D}$	Damping matrix
$\boldsymbol{f}(.)$	Vector function of its argument
$\boldsymbol{g}$	Vector of gravity torques
$\boldsymbol{q}$	Vector of joint positions
$\boldsymbol{x}$	Vector of Cartesian coordinates
$\boldsymbol{C}$	Coriolis and centrifugal matrix
$\boldsymbol{F}$	Vector of Cartesian forces
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{K}$	Stiffness matrix
$\boldsymbol{M}$	Inertia Matrix
$\boldsymbol{\omega}$	Vector of angular Cartesian velocities
$\boldsymbol{\tau}$	Vector of joint torques

List of Abbreviations

VIC	Variable Impedance Control
DS	Dynamical Systems
VSDS	Variable Stiffness Dynamical Systems
GMM	Gaussian Mixture Models
DOF	Degrees Of Freedom
RCC	Remote Center of Compliance
CNS	Central Nervous System
GMR	Gaussian Mixture Regression
SPD	Symmetric Positive Definite
DMP	Dynamic Movement Primitives
LfD	Learning from Demonstration
AA	Autonomous Agent
VF	Virtual Fixtures
UQ	Unit Quaternions
SEDS	Stable Estimator of Dynamical Systems

CHAPTER 1

Introduction

The incorporation of robots into our daily existence has been a visionary aspiration shared by authors, historians, filmmakers, and scientists alike. In the movie Intersellar for example, director Christopher Nolan envisioned a futuristic version of advanced robots, called TARS and CASE. The robots are portrayed to be highly intelligent entities, adept at taking decisions on their own, performing sophisticated tasks that demand superior reasoning capabilities and *physical interactions*, all while seamlessly *adapting* their behavior to suit the human preferences or to *comply* with their surroundings. Novelist Isaac Asimov on the other hand has long contemplated the principles that should govern human-robot co-existence. In his story Runaround, he identified *safety* as a core, indispensable value, epitomized through the famous Three Laws of Robotics:

1. "*A robot may not injure a human being or, through inaction, allow a human being to come to harm.*
2. *A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.*
3. *A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.*"

The examples mentioned above reflect a dream world humans have always yearned for: one where robots assist us in performing mundane, daunting, distant, and possibly dangerous tasks, and where humans and robots can harmoniously co-exist and collaborate together. With the advent of precise sensor technologies, efficient actuators, and computational power, robots are becoming increasingly popular nowadays, predominately in the industrial sector e.g the automotive industry. There, robots can perform a plethora of tasks at high speeds and formidable accuracy, often in deterministic, well-defined settings. Tasks such as picking and placing of objects,

palletizing, inspection and welding have been all automated to great effect thanks to robots.
Still, the longstanding vision for robots has always revolved around their transition from their well-defined industrial settings into more domestic and dynamic environments such as hospitals and restaurants, where they perform physical manipulation tasks and interact with humans. This vision inspired a new generation of torque-controlled robots with lightweight and compliant structures, with which a pool of prospects, but also challenges arise. While such robots clearly possess enhanced safety and performance features, without proper control, these robots offer no distinct advantages compared to their bulky, rigid counterparts commonly used in industry and their potential remains untapped. Indeed, it is crucial to design control algorithms that would enable these robots to interact in a *safe* and a *compliant* manner with their external surroundings, while being able to continuously *adapt* to dynamic and uncertain environments, in addition to fulfilling the task requirements. Until the day comes where true robot autonomy becomes a tangible reality, as it stands, it seems the only solution is for humans and robots to work together in some capacity to realize the human needs and demands. In this endeavor, teleoperation is one of the important tools humans devised to facilitate such cooperation. It lies in our nature to always conceive and innovate tools that would act as an extension of our arms. Consider cutting, for example, the knife extends our arms enabling us to cut objects that would be impossible to cut with our arms. In the same analogy, teleoperation serves as an extension of our presence into distant or hazardous domains, facilitated by robotic avatars capable of executing the task instead of us. This happens by having the human control a master interface, whose motion commands are transmitted into a remote robot[1]. Additionally, force feedback is transmitted from the remote robot back to the master, allowing the human to feel the physical interaction, as if (s)he is directly interacting with the environment.

This thesis is concerned with the development of control and learning techniques for robots during physical interactions, whether operating autonomously or in teleoperated settings. Central to this thesis is the concept of Variable Impedance Control (VIC) [1]. VIC is one of the main reasons why humans excel in performing complex manipulation tasks that require continuous adaptation to various task requirements and environmental conditions [2, 3]. In this thesis, we take inspiration from humans and leverage VIC as a methodology to shape the robot's interactive behavior during task execution. To this end, we exploit techniques from control theory, machine learning, and robotics. In the control part, our aim is to design VIC that can guarantee safety and stability during task execution, which is clearly of paramount importance if robots were to operate in dynamic environments, possibly populated by humans. On the other hand, we leverage learning for deriving suitable VIC policies from human demonstrations, which can subsequently be employed for robots.

[1]Historically, the remote robot was also called slave. However, the use of this word is being strongly discouraged in the robotics community nowadays

1.1 Contribution

Humans are characterized by a unique ability to skillfully perform tasks that feature physical interaction with their environments. Imagine cutting for example, we perform such a task with ease while exhibiting stable hand motions, despite the inherent instability arising from the interaction forces between the knife and the object being cut. Furthermore, we are able to optimize our effort thereby achieving efficient task execution, demonstrated by the fact that we automatically adjust our arm stiffness just as needed to achieve the task e.g we stiffen up our arm to cut a hard piece of meat, while being more compliant when cutting butter.
The capabilities exemplified above can be mainly attributed to our ability to regulate our arm endpoint mechanical impedance, which roughly speaking, refers to the degree our arm gives in or resists when subjected to a perturbation. In his famous trilogy [4–6], N. Hogan introduced impedance control as a general paradigm that can be used to control a robot behavior when physically coupled to its environment. Instead of conventional feedback control to minimize force or motion errors, the idea in impedance control is to modulate the dynamic relationship between the two, typically by a second-order dynamic model consisting of a mass, spring and a damper. Moreover, allowing the impedance parameters to vary over time can further improve the robot adaptation capabilities to different task contexts and environments. Since the 90's, numerous works have shown the benefits of VIC, in extending robot operation from simple motion tasks, to more complex interactions that can feature dynamic tool manipulations such as in valve turning [7], cutting [3, 8], and compliant grasping of fragile objects [9], or interactions with humans as in collaborative tasks [10,11]. The variation of the impedance parameters, however, comes at the cost of endangering the stability of the controlled robot. This problem is aggravated when taking into account robot redundancy. While redundancy enhances robot flexibility and dexterity by simultaneously executing several hierarchical task objectives, stability-violating control terms are introduced in the system [12]. Therefore, the first objective in this thesis is to design a variable impedance controller for multiple prioritized tasks, that not only guarantees the stability of the system, but also features an additional safety layer responsible of keeping the kinetic energy of the robot bounded below a safe limit.
In (variable) impedance control, the impedance behavior of the robot is specified with respect to an equilibrium trajectory- the trajectory the robot will follow if left unperturbed. Common trajectory representations such as splines are often parameterized with respect to time, with no regard to the actual robot state. Unfortunately, this can pose serious safety issues especially in uncertain and dynamic environments. Suppose that a mismatch between the desired robot task model and the actual environment would cause the robot to get stuck behind an obstacle while performing a reaching motion, the accumulated error between the actual robot pose and the desired one would lead to a large force due to the spring term of the robot impedance controller, leading eventually to damaging the robot or a violent abrupt robot motion if the obstacle is suddenly removed. To solve this problem, in this thesis, we propose a

control architecture where the current robot position is continuously fed back into a controller that concurrently regulates the robot motion and impedance, thereby avoiding the drawbacks of time-dependent trajectory parameterizations. The controller, termed Variable Stiffness Dynamical Systems (VSDS), combines the benefits of VIC, with Dynamical Systems (DS) motion generators [13]. Representing motion plans using DS is becoming increasingly popular thanks to their effectiveness in capturing a wide array of complex robot motions, their convergence properties to desired states and their flexibility for integration with various machine learning algorithms.
While the aforementioned contributions primarily revolve around autonomous robot operation, there still exist numerous scenarios where advanced human cognitive abilities and problem-solving skills remain indispensable. Bilateral teleoperation [14] emerges as a solution that harnesses these human capabilities, allowing robots to leverage human expertise for precise task execution in otherwise inaccessible environments e.g nuclear waste handling [15]. A recent trend in the robotics community has focused on transferring VIC methods into the realm of teleoperation, with the aim to endow robots with human-like adaptation capabilities during remote physical interactions. This trend has given rise to a new avenue of approaches termed "teleimpedance" [16], wherein both the human's motions and their estimated arm stiffness are conveyed in real-time to the remote robot. In this thesis, we take a novel stance on the teleimpedance problem. Instead of relying on additional sensors to measure the human arm stiffness, we employ Learning-from-Demonstration (LfD) to learn a variable stiffness control policy from human demonstrations, capable of shaping the remote robot impedance in a biomimetic manner.
Furthermore, we extend the proposed teleimpedance approach into a full shared control architecture dedicated to teleoperated contact tasks that require simultaneous control of motion, force and impedance. We also employ LfD to learn task models, which can be subsequently used to haptically guide the human along a nominal task trajectory that encodes the motion and force aspects of the task, while simultaneously adapting the remote robot impedance, based on the perceived state of the environment.

1.2 Outline

The work presented in this thesis is divided into 6 chapters. Figure 1.1 provides an overview of the contributions proposed in this thesis. In **Chapter 2**, we provide a concise overview of the relevant literature related to this thesis. This includes VIC for both humans and robots, the concepts of teleoperation and shared control, as well as safety-aware robot control. Moreover, the chapter outlines the core theoretical concepts that serve as the foundation for this thesis. These encompass robot modeling and impedance control, stability theory, and various machine learning regression techniques.
Following this background chapter, the main contributions of this thesis are presented. In **Chapter 3**, we present a hierarchical passivity-based variable impedance controller. First, the controller is designed for the simple case where the hierarchy consists of two

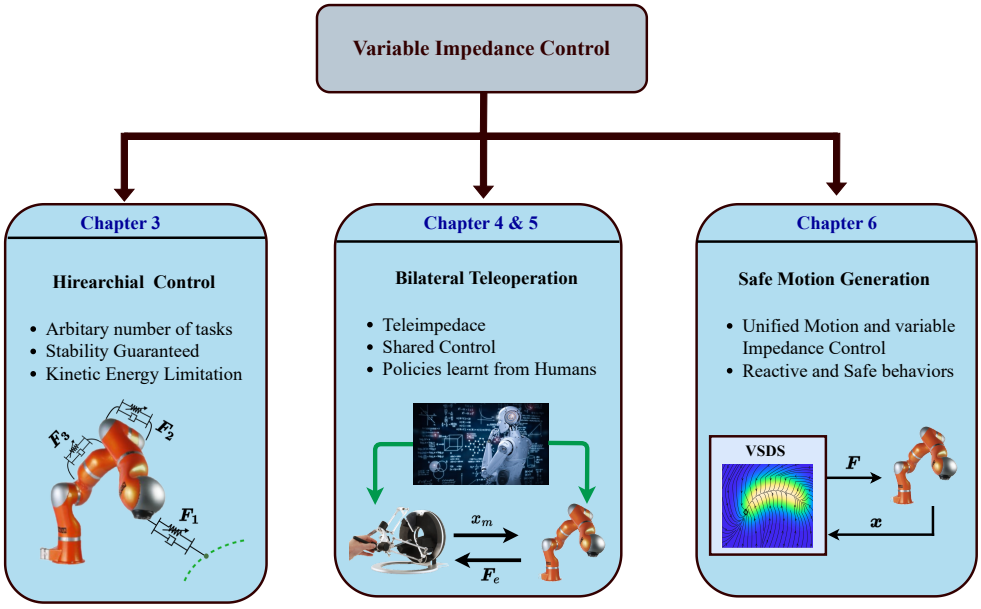


Fig. 1.1: Conceptual Illustration of the main contributions presented in this thesis. These contributions are anchored within the Variable Impedance Control framework and revolve around three main axes: i) Hierarchical control for redundant manipulators, ii) bilateral teleoperation, and iii) safe motion generation and control.

tasks prioritized in a strict manner. Then, the controller is extended to the case where we have an arbitrary number of prioritized tasks. By ensuring passivity, we guarantee the stable interaction of the robot with any passive environment. Furthermore, we enhance the safety of the robot by limiting the maximum kinetic energy the robot can generate while moving. The controller is validated in simulations and on real-robot hardware.

Chapter 4 and **Chapter 5** are concerned with teleoperation. In **Chapter 4**, we propose a variable impedance teleoperation architecture, which does not require an explicit impedance command channel. Instead, we learn a stiffness adaptation policy from human demonstrations, encoded in the form of Gaussian Mixture Models (GMM). Furthermore, a passification layer is designed to guarantee the stability of the closed-loop system. The presented approach is validated in a teleoperated cutting task, under various conditions, showing superiority in performance with respect to a constant stiffness setting for the remote robot.

In **Chapter 5**, a shared control architecture that builds on the teleimpedance approach from chapter 4 is presented. GMM are also employed to learn task models consisting of motion, force and stiffness profiles. While the stiffness profile is used to shape the remote robot impedance, the learned motion and force models serve as a reference to a virtual fixutre that guides the user along a trajectory capturing the nominal task dynamics. The approach is validated in a user study on two contact tasks: drawing and wiping.

Chapter 6 tackles the problem of safe motion generation and control. Herein, we present a controller capable of shaping the robot behavior according to a desired stiffness profile, as well as a reference robot motion represented as a first-order DS. The benefits of the controller are showcased in autonomous task execution, and also in a shared control setting. The controller is then extended to exhibit features such as passivity, assymptotic stability and reference velocity tracking. Additionally, the controller is formulated for the orientation case, using a unit quaternion representation and tools from Lie theory.
Finally, **Chapter 7** concludes this thesis highlighting the main contributions and lessons learned in this thesis. The chapter also provides recommendations for future work directions and extensions of the work presented in this thesis.

1.3 Publication Note

Most of the contents of this thesis have been published in peer-reviewed conferences and journals. The publications that this thesis is based on, are enumerated below:

Journals:

1. **Y. Michel**, C. Ott and D. Lee, "Safety-Aware Hierarchical Passivity-Based Variable Compliance Control for Redundant Manipulators," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3899-3916, 2022, [17].

2. **Y. Michel**, R. Rahal, C. Pacchierotti, P. R. Giordano and D. Lee, "Bilateral Teleoperation With Adaptive Impedance Control for Contact Tasks," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5429-5436, 2021, [18].

3. **Y. Michel**, Z. Lee and D. Lee, "A Learning-Based Shared Control Approach for Contact Tasks," in *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8002-8009, 2023, [19].

4. **Y. Michel**, M. Saveriano , D. Lee, "A Passivity-Based Approach For Variable Stiffness Control with Dynamical Systems". *IEEE Transactions on Automation Science and Engineering*, 2023, [20].

5. H. Xue[†],**Y. Michel**[†], and D. Lee, "A shared control approach based on first-order dynamical systems and closed-loop variable stiffness control," *Journal of Intelligent and Robotic Systems*, vol. 109, no. 4, pp. 85-98, 2023, [21].

[†]Equal Contribution

Conferences:

6. **Y. Michel**, C. Ott, and D. Lee, "Passivity-based variable impedance control for redundant manipulators," *IFAC-PapersOnLine*, pp. 9865-9872, 2020, [22].

7. X. Chen†, **Y. Michel**†, and D. Lee, "Closed-Loop Variable Stiffness Control of Dynamical Systems." *IEEE-RAS 20th International Conference on Humanoid Robots*, pp. 163-169, 2021, [23].

8. **Y. Michel**, M. Saveriano, F. J. Abu-Dakka, D. Lee, "Orientation Control with Variable Stiffness Dynamical Systems," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4457-4463, 2023, [24].

The author of this thesis also contributed to the following publications, which however were not included in this thesis:

9. **Y. Michel**, M. Saveriano , D. Lee, "A Novel Safety-Aware Energy Tank Formulation Based on Control Barrier Functions", *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5206-5213, 2024, [25].

10. **Y. Michel**†, Y. Abdelhalem† , G. Cheng, "Passivity-Based Teleoperation with Variable Rotational Impedance Control", *IEEE Robotics and Automation Letters*, (under review), 2024, [26].

11. R. Arduini†, **Y. Michel**†, H. Singh, J. Klodmann, and D. Lee, "Learning From Demonstration of Robot Motions And Stiffness Behaviors For Surgical Blunt Dissection," *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*,(accepted), 2024.

12. **Y. Michel**†, K. H. Schulleri†, L. Johannsen, and D. Lee, "Coordination tending towards an anti-phase relationship determines greater sway reduction during entrainment with a simulated partner," *Human Movement Science*, vol. 89, p. 103090, 2023, [27].

13. H. Ahn, **Y. Michel**†, T. Eiband†, and D. Lee, "Vision-based approximate estimation of muscle activation patterns for tele-impedance," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5220-5227, 2023, [28].

14. R. Arduini, **Y. Michel**, H. Singh, J. Klodmann, and D. Lee, "Autonomous Skill Transfer for Blunt Dissection: Learning Stiffness and Motion Profiles from Data," *International Workshop on Data vs Model in Medical Robotics at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

15. K. H. Schulleri, L. Johannsen, **Y. Michel**, and D. Lee, "Sex differences in the association of postural control with indirect measures of body representations," *Scientific Reports*, vol. 12, no. 1, 2022, [29].

16. K. H. Schulleri, **Y. Michel**., D. Lee, and L.Johannsen, "High-order body representations affect human body sway." *65th Annual Meeting of the German Society for Clinical Neurophysiology and Functional Imaging (DGKN)*. 2021.

17. I. Tannert, K. H. Schulleri, **Y. Michel**, S. Villa, L. Johannsen, J. Hermsdorfer, and D. Lee, "Immediate effects of vibrotactile biofeedback instructions on human postural control," *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2021, [30].

CHAPTER 2

Background

In this chapter, we provide a brief overview on some of the relevant literature and technical preliminaries this thesis is built on. In the first part of the chapter, we review works related to VIC in humans and robots, Shared Control for telerobotics, and Energy-Aware control. Then, we elaborate on the main theoretical concepts used in this thesis, which include robot modeling and impedance control, stability theory, and some regression techniques.

2.1 Related Works

2.1.1 Interaction Control

Early works in manipulator control have primarily focused on developing controllers aimed at enabling a robot to closely track a desired trajectory [31–33] while considering potential model uncertainties [34] or robot flexibility [35]. This requirement is fundamental for several applications, particularly in the industrial sector, where robots are expected to perform predefined motions with precision and reliability. In tasks such as pick-and-place or spray-painting, robots operate in free motion, making this control approach sufficient. However, numerous tasks, such as wiping or screwing, necessitate physical interaction between the robot and its environment. Using a stiff motion control approach in such cases can result in excessive contact forces, potentially causing damage to the robot or the surrounding environment. To address this issue, it is essential to design controllers that explicitly shape the robot's physical behavior during interaction. In this spirit, the Hybrid position-force control framework was introduced [36, 37], drawing inspiration from the observation that many tasks such as wiping, impose kinematic constraints on the robot. Specifically, certain Degrees Of Freedom (DOF) may only allow force application, restricting motion. As a result, the controlled space of the robot can be divided into two sub-

spaces: motion and force-controlled subspaces. To determine which directions fall under motion or force control, a binary selection matrix is specified. These directions are defined relative to a task frame, which establishes the axes along and around which the task constraints are defined. In [38], the operational space formulation was presented in order to enable hybrid motion-force control, with a complete dynamic compensation of the manipulator dynamics expressed in the end-effector frame, where the task goal and geometry are typically specified. This eliminates dynamic couplings between different task directions, which allows to perform fast motions with large accelerations.

A fundamental requirement in the above formulations is to accurately obtain a geometric model for the task in the form of a task frame and the selection matrix. In practice, planning errors or unexpected impacts can always be encountered, and therefore, a control strategy must be robust enough to adjust to such situations. The parallel force-position control approach [39] was developed in order to endow a robot with such robustness, where force and position control are enabled simultaneously in all Cartesian directions. The design choice was to provide dominance for the force control loop via a Proportional Integral (PI) control action, which allows to effectively handle deviations from the planned task.

A different perspective on the force control problem was taken in [40], where the author introduced the stiffness control approach. It allows the designer to simultaneously control positions and forces by specifying the nominal position of the robot as well as a stiffness matrix. A somewhat similar idea was proposed in [41], where the stiffness of the robot is varied in realtime in order to achieve force tracking, without requiring any knowledge about the environment.

The approaches in [40] and [41] can be seen as a form of Impedance Control, which was introduced in the seminal work of N. Hogan [4–6]. The main motivation for impedance control is that during physical interactions, the manipulator is coupled to its environment and therefore, can be no longer treated as an isolated system. This means that for the general case of dynamic interactions, where the work done by the robot on the environment is not zero, control of either forces or positions is not well-posed, since these quantities depend on both the robot and the environment, and therefore, in principle cannot be determined by either entity. On the other hand, we can specify the "disturbance response" of the robot with respect to a virtual desired equilibrium, and this would remain an exclusive property of the robot. Therefore, the goal in impedance control is to regulate the energy exchange between the robot and its environment by assigning a prescribed target behavior, for example in the form of second-order dynamics consisting of a mass, spring and a damper. In this spirit, impedance control should be viewed as a control philosophy rather than a particular control implementation. This also provides the freedom in implementing the target behavior, while exploiting the available software or hardware resources. For example, it is possible to obtain impedance control passively, by exploiting inherent compliance in the robot hardware. Industrial applications that feature contact with the environment, as in peg-in-hole assembly operations, often employ a mechanism called the Remote Center of Compliance (RCC) [42]; a mechanical device attached

to the robot end-effector to ensure a compliant behavior during interactions. Another possibility are robots that employ flexibility in the links [43], or even in the actuation [44]. Alternatively, active impedance control is implemented via software through a control algorithm purposefully designed to achieve the desired target behavior. Depending on the control interface available to the robot i.e torque or position control, the causality of the controller is determined. If the robot is torque-controlled, then the controller admits an impedance causality where the controller takes as input a velocity, and outputs a force in response. On the other hand, in position-controlled robots, as typically the case in industrial manipulators, the desired dynamics are simulated to obtain a position in response to an input force. Then, an inner stiff position control loop is used to accurately track the simulated position, in order to recreate the target behavior. In general, it is well known that impedance control [45] performs well during contacts with stiff environments, but rather poorly during free motions. On the other hand, admittance control provides high accuracy during free motion, but can suffer from instablities during physical interactions. A hybrid impedance-admittance controller that combines the benefit of the two approaches was proposed in [45].

2.1.2 Variable Impedance Control

VIC in humans

Humans exhibit an amazing capability to proficiently execute tasks demanding physical interactions with their environment. Many of these tasks, such as those that involve tool use, are inherently unstable. Still, we gracefully perform these tasks in a stable manner, while also optimizing our effort in order to achieve the task goal in the most efficient way possible. The remarkable skills humans demonstrate during physical interactions have been long studied over the past years, with the aim to understand the underlying cognitive, sensory, and motor mechanisms governing these interactions. It was established very early on e.g [46], [47], that our arm exhibits predominantly a spring-like behavior when disturbed from an equilibrium position, due to the stretch reflex and the elastic muscle properties. Changes in arm configuration, as well as the coordinated action of agonist and antagonist muscles, influence the magnitude, shape and orientation of the stiffness ellipsoid at the arm endpoint [48]. Beyond the static aspect, we are also able to control the mechanical impedance [49], by changing the apparent inertia of our arm, as well as adjusting our arm visco-elastic properties (i.e stiffness and damping) in a task-dependent manner [50].
What strategies do humans adopt during physical interactions? First, it should be noted that controlling the mechanical impedance of our arm constitutes only one aspect of the control strategy, namely, the feedback control part. The long latencies in our Central Nervous System (CNS) mean relying solely on a pure feedback control strategy is prone to failure due to potential instabilities [51]. Therefore, the CNS employs a feedforward mechanism that anticipates and plans in advance the forces necessary to achieve a task, based on the estimated internal and external dynamic

effects [52, 53]. Previous works have shown that when subjected to stable velocity-dependent force fields in planar reaching motions, the CNS adapts the feedforward forces to compensate for the environmental disturbance and reduce motion error [54]. On the other hand, coping with unpredictable dynamics or environmental instabilities typical of tool-use [55], requires an increase of the mechanical impedance to maintain a stability margin [2]. In a series of studies [56–58], it was shown that when subjected to novel dynamics, the early stages of learning feature an increase in the arm stiffness to guarantee stability. As learning progresses and an internal model of the task dynamics is developed, the arm stiffness gradually decreases. In a nutshell, the CNS learns to interact with environments that exhibit both stable and unstable dynamics by continuously adapting the endpoint feedforward force and impedance, through the concurrent minimization of both movement errors and efforts resulting in optimized interaction strategies. This process can be formalized using non-linear adaptive control techniques [59, 60], which can be useful to predict human behaviors and potentially, develop control strategies that enhance the performance and efficiency of robots, as will be discussed next.

VIC in robots

Since their conception, the idea for robots has long been centered on their evolution beyond industrial spaces, into more domestic and dynamic settings. This encouraged roboticsts to seek advanced methodologies to improve existing robot technology, both on software and hardware levels. In this regard, understanding how humans employ VIC to realize their unique physical interaction skills has inspired research aimed towards bringing VIC to robots. It wasn't long after Hogan proposed impedance control as a paradigm to controlling a robot's physical interactive behavior, that researchers started exploring the possibility of varying the robot impedance parameters in a myriad of applications and tasks, ranging from rehabilitation [61, 62], surgical robotics [63, 64], grasping [9] and human-robot collaboration [65].
VIC can be realized both actively via software [1], and/or passively in hardware [44]. In the latter, elastic elements such as springs and dampers are embedded in the robot actuators, in order to obtain inherent constant or variable compliance. This equips robots with additional features such as enhanced safety and energy efficiency, in terms of generating cyclic or explosive motions such as throwing [66] or kicking [67]. Notable examples for actuators with inherent compliance are the Series Elastic Actuators [68], where a constant spring is placed in series with the actuator, and Variable Stiffness Actuators [69, 70], used for example in the DLR-hand arm system David [71], and where the spring stiffness is varied e.g by changing the spring preload. On the other hand, in this thesis, we mainly focus on realizing variable impedance behaviors actively via software. Two central questions arise in this context: firstly, ensuring the stability of variable impedance control policies, which will be further discussed in the subsequent section, and secondly, identifying variable impedance profiles suited for the given task or performance objectives.
In this regard, several works devise such profiles by handcrafting adaptation policies, which typically requires expert knowledge of the task requirements and environment

conditions, as well as the available robot hardware. Ikeura and Inooka [72] were the first to propose VIC as a method for cooperative manipulation tasks, where the robot is controlled in low or high damping modes, depending on the velocity of the human. Later, the authors extended their framework to find the optimal damping factor by minimizing a suitable cost function [73]. In [74], the authors also vary the damping, however, based on an estimate of the human arm stiffness computed via differential changes of force and position. The idea of using information about human arm stiffness estimates to shape the robot variable impedance behavior was also pursued in works such as [75] and [76]. In rehabilitation, the work in [61] adapts the impedance of an ankle-foot orthoses during the walking cycle in order to treat a drop-foot type of motor deficiency. The authors propose a stiffness profile that varies such that movements that do not align with the pre-defined motion plan are discouraged, while also realizing an "assist-as-needed" interaction.
The impedance adaptation policies in the aforementioned works are limited to their particular application field and therefore have a restricted ability for generalization. Furthermore, as previously indicated, the formulation of such policies necessitates profound domain knowledge which clearly limits their application in other contexts. To solve this, several works have started exploring machine learning solutions to the VIC problem, whether in supervised or unsupervised settings. Traditionally, imitation learning has been concerned with learning motion trajectories [77]. More recently however, learning variable impedance profiles from demonstrations has also received significant attention. For example, Kormushev et al. [78] learn motion and force profiles via Gaussian Mixture Models (GMM), while the stiffness is obtained via least-square fitting of the residuals from the regression process. A similar idea was proposed in [79], where the stiffness is learned also via least-squares fitting, while the corresponding attractor trajectory is encoded via Task-parameterized GMM. The authors validate their approach in collaborative tasks of table transportation and furniture assembly, where the robot displays a compliant behavior according to the perceived human intentions. The authors also explored in [80] formulating the problem using Optimal Control, solved using an infinite horizon Linear Quadratic Regulator (LQR). In [7], GMM model the relationship between the stiffness variations and the sensed external forces, which according to the authors, capture information that can be useful to shape the impedance behavior of the robot e.g friction level of a valve. In addition to the standard Cholesky decomposition, the authors also explore GMM and Gaussian Mixture Regression (GMR) for Riemanian manifolds [81], since such a formulation respects the underlying geometry of Symmetric Positive Definite (SPD) matrices. In a subsequent work [82], the authors reformulated the well-known Dynamic Movement Primitives (DMP) [83] to work with SPD data, using Riemanian operations.
The above works do not measure the stiffness information explicitly during demonstrations. Instead, they assume that the stiffness can be somehow extracted from the observed task dynamics. Alternatively, it is possible to rely on an external interface as an explicit stiffness command channel. Kronander et.al. [84] proposed a method to enable a human teacher communicate to the robot stiffness information by physical

interaction, making use of robotic skin. In [85] and [86], the authors use the interface developed in [87] to learn motion and stiffness profiles using a bilateral teleoperation setup and Electromyography (EMG) to convey the desired stiffness. The use of EMG for stiffness learning was also explored in other works such as [88], [89] and [90].
From a different perspective, variable impedance profiles can be obtained as a bi-product to the learning process of other relevant aspects of the robot control policy. For instance, works such as [91] and [92] use GMM to encode the motion trajectories of a given task, while the stiffness is shaped to be inversely proportional to the covariance information of the GMM. The reasoning for such a setting is that areas with low variance indicate high certainty about the desired trajectory. Therefore, the robot should be stiff to improve tracking performance in those areas, and compliant otherwise. In [93], the robot's stiffness is also high close to the reference trajectory, and decreases smoothly whenever the human applies an external force that indicates a corrective action. Similarly, The progressive automation framewok [94, 95] aimed at the incremental learning of periodic tasks, relies on a stiffness adaptation policy shaped by the trajectory tracking error, as well as the external forces of the human, such that the robot smoothly shifts its role from leader to follower, and vice-versa. Passivity is also guaranteed through the use of energy tanks [96].
A major limitation in imitation learning frameworks is that the learnt policy is only as good as the demonstrations provided by the human. Moreover, providing demonstrations can be difficult for complex tasks such as flipping a pancake [97]. Such shortcomings can be solved by iterative and reinforcement learning techniques where the robot learns the optimal policy on its own. In a series of studies [3, 8, 98, 99], the authors formulated an iterative learning control law inspired by human motor control principles. The proposed approach relies on the concurrent adaptation of feed-forward force, impedance, and reference trajectory in order to minimize instability and effort, by optimizing a suitable cost function. The controller was later utilized in a collaborative task setting [100], where an adaptation law was additionally proposed for the rendered robot inertia, and where the robot desired trajectory is encoded with Dynamical Systems (DS). In [101], the adaptive controller is central to a framework dedicated to formalizing and learning contact-sensitive tasks such as peg-in-hole assembly. Along the same lines of [98], an online version of the Expectation maximization algorithm [102] is used in [103] for the incremental learning of feedforward force and stiffness to compensate for disturbances during dynamic contact tasks.
Closely related to iterative learning, Reinforcement Learning has also been employed to great effect in VIC. In [104], the Natural Actor-Critic algorithm from [105] is used to learn a two-dimensional stiffness matrix parameterized by three parameters, in order to keep the search space limited. Buchli et al. [106] on the other hand, use Policy Improvement with Path Integrals [107] to learn a control policy represented as a DMP that encodes a reference motion and a diagonal stiffness profile. The used cost function is inspired by human motor control theory, and formulated to minimize tracking errors and control efforts. The idea is further extended in [108] to learn full stiffness matrices with non-zero coupling terms, which allows to also learn the synergies between the motion directions. In [109], the authors represent the policy

with first order DS, which in contrast to DMP, eliminates the time dependency and shows robustness to temporal perturbations. Recently, to guarantee safe explorations, the work in [110] exploits the IMOGIC formulation of [111] to represent the controller in order to learn inherently stable policies.

Stability in VIC

Unfortunately, the variation of the impedance parameters could potentially result in violating the stability of the controlled robot. This could be a problem, especially if robots are envisioned to work in close proximity with humans. Therefore, several works have focused on devising techniques to guarantee the stability of a variable impedance control law. The human-like adaptive impedance controller from [98, 99] derives conditions on a Lyapunov function to guarantee the stabilization of the trajectory tracking errors, as well as the errors between the actual and estimated impedance parameters of the environment. From a different perspective, the use of *passivity theory* [112] provides a powerful tool to analyze the interaction of a robot with its environment, since it does not assume any knowledge about the environment other than it being passive. In this regard, Ferraguti et.al. [96] proposed the use of energy tanks to harvest the energy dissipated in the system, and compensate for the non-passive control actions. The approach was later extended in [113] where the time-varying admittance case was considered, from a port-Hamiltonian modeling perspective. Energy tanks however need proper initialization and otherwise, can result in an unexpected online modification of the impedance parameters. This was argued in [114], where the authors borrowed techniques from adaptive control to derive state-independent conditions on a variable compliance profile (i.e stiffness and damping). This permits the designer to know a priori the feasibility of commanding a certain desired impedance to the robot. A similar idea was adopted in [115], where using Linear-Time-Varying system formulation, the authors are able to derive conditions on the impedance parameters to guarantee exponential stability. The Passivity-Preservation Control (PPC) scheme was presented in [116], in order to guarantee passivity despite impedance variations. The PPC framework is tailored to suit robots with flexible joints, since the presented stability analysis considers both time-varying active and passive components of the stiffness. The resulting control is a full-state feedback that takes into account motor and link-side variables. Finally, considering redundancy, the authors in [117] exploited robot redundancy to decouple the apparent inertia, thereby enlarging the stability region in the impedance parameter space, where the region was determined experimentally.

Teleimpedance

So far, the reviewed literature in VIC has primarily focused on scenarios where the robot operates autonomously. In many cases, however, human cognitive abilities and problem solving techniques are still required, especially to perform critical tasks in inaccessible environments. This is typically solved by teleoperation, where the commands of the human operator are transferred to a remote robot through a suitable

interface (master). In traditional teleoperation [14], the remote robot is programmed to be stiff in order to follow the motion of the human operator as accurately as possible. This, however, can result in high interaction forces, which might damage the environment and further aggravates instabilities in the system, especially in bilateral teleoperation, where force feedback from the remote site is transmitted back to the human. Furthermore, as stated earlier, such a strategy is inconsistent with humans' strategy during physical interactions, which is to continuously adapt their endpoint impedance depending on the environment and the task requirements. Inspired by these issues, the concept of teleimpedance was proposed, where the basic idea is that in addition to the reference motion, the human commands the impedance to the remote robot. Obviously, this requires an interface that enables the human to modulate the desired impedance of the remote robot. Early works in this regard relied on grip force sensing [118], motivated by the fact that the grip force highly correlates with the human arm stiffness [119]. This idea was also used where user inputs modulate the stiffness of a remote robot with variable impedance actuation [120]. The low-frequency modulation was controlled via software, while high-frequency impedance was regulated via the actuator. These approaches, however, can only command one DOF stiffness input, which means that the approach is not suitable for complex interaction tasks that might require stiffness modulation in multiple axes, as in peg-in-hole tasks. This was solved in [16], where the term "teleimpedance" was first proposed. The authors rely on the use of EMG, a kinematic model of the human arm, and offline calibration in order to estimate in real time the stiffness of the human arm endpoint and command it to the robot. The proposed setting allows the human to command both the orientation and magnitude of the stiffness ellipsoid, as well as the reference motion of the arm obtained via a Motion-capture system. From thereon, several approaches that build on the original EMG-based teleimpedance concept were proposed. For example, Laghi et al. [121] incorporated the idea in a conventional bilateral teleoperation architecture with force feedback where passivity was also guaranteed using the Time Domain Passivity Approach [122, 123], while Peternel et al. [11] simplified the setup for estimating a one DOF stiffness trend. In [87], Yang et al. also used muscle activations for commanding the remote robot stiffness in a conventional teleoperation setting. The proposed setup is meant as a skill transfer interface that enables a human to teach a robot motion and impedance profiles for task execution.

Relying on sensing the grip force or the muscle activation typically requires a calibration procedure, and therefore, these approaches are rather person-specific. Furthermore, the measurements might be contaminated by other factors, such as fatigue. This was argued by [124], where inspired by early works on human arm stiffness estimation approaches, the authors proposed to introduce perturbations to the master robot and estimate the human arm impedance by measuring the deviations to the nominal task trajectory and subsequently regression. This was used in the context of human assistance where perturbations are frequent.

The aforementioned approaches rely on sensing one way or another the natural impedance of the human arm. A different perspective to teleimpedance is to assume

that the human can convey to the robot the adequate impedance necessary to achieve a given task. The work in [125] proposed a joystick-like device that allows the human to manually control the remote robot stiffness. On the other hand, in [126], the authors use voice commands to make the robot stiffen up or become more compliant. Recently, the work in [28] adopted an interesting approach to the teleimpedance problem, where vision-based teleimpedance was explored. The presented framework uses deep learning to map forearm images captured from a commercial webcam, to muscle activation patterns, from which the robot stiffness profile can be reconstructed. EMG sensors are only needed during the training stage, while during execution, the trained model operates solely based on the captured images, and is able to command the remote robot stiffness in real time.

2.1.3 Feedback Motion Planning

The requirement for safe and compliant control should also be complemented with a motion generator that provides the reference path or trajectory for the controller to track. Over the past decade, numerous solutions have been proposed in the literature to endow low-level control with motion generators that can guarantee flexibility and precision, while possibly taking into account constraints such as obstacle avoidance or actuation limits. Worth noting in particular the Learning from Demonstration (LfD) paradigm [77], which recently received a lot of attention in the robotics community, especially with the increasing popularity of machine learning algorithms. Therein, demonstrations provided by the human via teleoperation or kinesthetic teaching are encoded via regression models that can be subsequently employed for realtime motion generation. Techniques such as GMM [78, 127], Hidden Markov models [128, 129], Gaussian Processes [130], DMP [83] and first order DS [13] have all been successfully deployed to modeling a wide range of complex robot motions.
Unfortunately, the classical way to command desired motions is to program the impedance controller and motion generator as two separate loops, where time-indexed trajectory generators such as splines or DMP feed the controller with a sequence of desired setpoints parameterized with time. This is an open-loop configuration, where the motion generator does not have any feedback on the current robot state, and therefore clearly lacks robustness to temporal perturbations [131]. Furthermore, this raises major safety concerns in unstructured or populated environments where potential unplanned collisions would lead to very high forces or clamping situations that pose serious dangers to nearby humans, or to the damage of the robot or its environment [132].
To tackle this problem, the concept of *Feedback Motion Planning*[1] aims to equip robots with reactivity that allows for realtime adaptation to dynamically changing environments. As the name suggests, the main idea is to continuously feedback the current robot state into the motion generator, such that the desired motion is

[1]Strongly related to feedback motion planning is the concept of optimal feedback control, which aims to generate optimal control commands given a cost function. On the other hand, in feedback motion planning, the focus is specifically on the motion generation problem.

commanded in a robot-aware manner. Two categories can be distinguished in this context: planning and control level reactivity. In the former, the planner and the controller still operate in two separate loops, however the generator replans and adapts the desired path or trajectory in realtime, based on the measured robot state. For example, [133] presents a closed-loop version of the well-known Rapidly exploring Random Trees (RRT), where the "tree" describing the current desired motion plan is consistently re-propagated based on the robot state and the environmental constraints. Alternatively, some works [134, 135] use the notion of a funnel; an attractive region with a unique goal position, that can viewed as an asymptotically stable dynamical system. By using a single or a composition of overlapping funnels, it is possible to drive the robot from anywhere in the configuration space to the final goal position. A different perspective to the feedback motion planning problem is to achieve reactivity on the control level, by integrating motion generation and control in a unified block, such that no distinction between the two is made. A single control policy is responsible for shaping the robot motion as well as simultaneously driving the robot along the desired trajectory, and possibly regulating the robot behavior during physical interaction. The seminal work of Khatib [136] introduced the well known concept of a potential field, where the robot control law is computed as the negative gradient of a potential function. The structure of the potential field can be tailored to encode attractive behaviors that drive the robot to the goal location, or repulsive behaviors to avoid obstacles. The work in [137] extends the capabilities of classical potential fields and proposes an approach to learn from human demonstrations non-parametric potential functions together with a dissipative field that encode dynamic and compliant behaviors. A similar objective was pursued in [111], where the motion generation and impedance control are combined through a GMM formulation into a single loop regulating the robot motion and interaction behavior simultaneously, while ensuring stability. Also using learning, the authors in [138] proposed an approach based on Gaussian processes to learn a combination of stiffness and attractors, in an interactive manner via corrective inputs from the human teacher.

Closely related to potential fields, several works have advocated for the use of closed-loop velocity fields to encode motion tasks. This is especially relevant for contour tracking tasks, where the timing aspect is not important. Instead, it is crucial that the robot stays on the path at all times. In fact, using a time-indexed trajectory tracking controller in such situations can be detrimental to performance, since the robot might temporarily leave the contour in order to catch up with the desired trajectory. The work in [139] proposes a feedback control law to drive the robot along the integral curves of a velocity field, augmented with a virtual flywheel capable of storing and releasing energy, in order to ensure passivity in the system. In [140], the authors realize the same goal by designing the controller as an interconnection of port-Hamiltonian systems and therefore guarantee passivity. Power-continuous control terms reroute kinetic energy flows in the system into desired and undesired directions in order to achieve asymptotic convergence to the contour.

2.1.4 Energy-Aware Control

Energy is one of the most fundamental concepts in science and engineering, and can be considered *la lingua franca* across the various physical domains. Energy and power considerations have been central to modeling and control design over the past years. The port-Hamiltonian framework [141] models physical systems as entities that exchange and route energy, through so-called *power-ports*, and where physical properties such as energy storage and dissipation of free energy in the system are made easily apparent. Closely related, graphical modeling techniques such as bond graphs [142] visualize these properties in the system, highlighting energy transformation and power flows between the constituting components of the system. In control, energy plays a crucial role in a branch of non-linear control theory called passivity-based control [143], which aims at designing controllers that guarantee the passivity of the controlled system. This ensures stability by making sure that the energy of the controlled system remains bounded, such that no internal generation of energy occurs. This methodology found its usefulness in various applications ranging from the control of robotic manipulators [31], legged [144] and humanoid [145] robots, and haptic interfaces [122]. Techniques such as energy shaping [146], interconnection-damping assignment [147] and the Time-domain passivity approach [122] are few examples among the numerous passivity-based controllers developed over the years. The energy-shaping technique was central in one of the most popular approaches to motion control of robotic manipulators: the PD+gravity compensation [31].
It is evident that such an energy perspective to control design, especially in the context of robotics, enriches a controller with a nice physical interpretation. In fact, in the *control by interconnection* paradigm [148], controllers are no longer viewed as signal processors, but rather as virtual physical systems with their own dynamics, power ports, and energy function. Ideally, the controller is designed to be passive, such that when it is interconnected to a (passive) plant, in a power-preserving manner, the controlled system (remains) is passive. Additionally, unwanted properties of the original plant are eliminated, in favor of new desired performance objectives for the closed-loop system e.g a new energy function describing a spring potential.

Energy tanks

For a robot that physically interacts with its environment, guaranteeing a stable behavior is of paramount importance both for safety and performance reasons. Clearly, this physical interaction, by definition, will involve the bilateral exchange of energy. In this regard, passivity theory offers an intuitive and a powerful tool to analyze the stability of the robot when interacting with possibly unkown environments, whithout making assumptions about the dynamic model of the environment e.g linearity. In fact, as proven in [149], it is possible to characterize a passive environment that when interconnected to a non-passive robot, the system becomes unstable. In this view, to ensure passivity, extending the system with so-called energy tanks is a well-established tool by now and has featured prominently in many robotic applications. Roots to the idea of energy tanks can be traced back to [140], where the principle of energy routing

was proposed, and whereby, it is possible to route and direct energy flows between systems by designing non-trivial power-preserving interconnections. Fundamentally, an energy tank is a virtual storage element, interconnected to the system in a power preserving manner, and where by manipulating the interconnection structure it is possible, to passify any control action. In this regard, the tank should be viewed as a safety mechanism that monitors the consequence of non-passive control actions, ensuring that the energy these actions inject into the system remains bounded.
As mentioned earlier, works such as [96] and [113] employed energy tanks to ensure passivity in the case the interaction behavior of the robot is specified with a time-varying stiffness and admittance, respectively. In [150], a two-layer approach for delayed bilateral telemanipulation is proposed. While the "performance layer" provides transparency, the safety layer ensures passivity by augmenting the master and remote robots controllers with two energy tanks. Energy tanks were also utilized in the shared control domain to passively implement virtual fixtures or haptic guidance forces [151]. The authors of [131] use energy tanks to ensure passivity when the robot is following the integral curves of a non-conservative first order DS. The problem of passivity loss due to force control actions is treated with energy tanks in [152] for a flexible joint robot, and for a hexa-rotor in [153]. In [154], energy tanks are used to preserve the non-passive null space projections for a two-level hierarchy. The authors extended their formulation in [12] for an arbitrary number of prioritized tasks.

Energy-based safety control

While guaranteeing a stable behavior is an important requirement for safety, defining concrete safety metrics and incorporating them into the controller design is one step further to enhance robot safety and reduce possible risk of injuries. Complementary to passivity and stability, the use of energy can also be applied for safety considerations and analysis of a controlled robot. This is also reflected in ISO safety norms such as the Speed and Separation Monitoring (SSM) and, Power and Force Limiting (PFL) which in essence limit the robot kinetic energy depending on the distance between the human and the robot, or such that a contact does not lead to injuries [155]. The works in [156], [157] identified the minimum amount of energy that can cause cranial bone failure and neck fracture, in clamped and unclamped situations. Inspired by that, [158] proposed an impedance controller that takes into account energy and power-based safety metrics, scaling the stiffness to regulate the energy, while damping is injected to confer with the defined constraints. Additionally, energy tanks are also used to passively implement the stiffness variations.
Along the same lines, [159] modeled a human-robot collaboration setup using the port-hamiltonian framework, and accordingly derived appropriate scalings for the robot stiffness and damping to limit the maximum energy and power that can be transferred to the human during human-robot object co-manipulation. While these works scale the robot impedance parameters, in [160], the energy in the system is regulated to remain below a safe threshold via scaling the desired trajectory of a robot with a passive compliant actuator. Similarly, [161] modified the desired velocity of an impedance-controlled robot to limit the maximum energy transferred

to a human in a rehabilitation setup. The energy limit was adapted in order to provide subjects an "assist-as-needed" rehabilitation experience. Similar ideas were also proposed in [162] and [163], in an iterative learning setting of periodic motions in a hybrid motion-force control task. The authors later extended their work to a general framework that limits the maximum energy and power transferred from an energy tank to a controlled robot [164, 165].

2.1.5 Teleoperation and Shared Control

Teleoperation: An historical perspective

As pointed out earlier, teleoperation offers a viable solution in several settings, where fully autonomous robot operation might not be possible. The concept of teleoperation was first proposed by Raymond C. Goertz [166] to enable material handling in nuclear waste sites. Initially, the teleoperation system was built as mechanically coupled linkages, allowing the human to transmit motions and receive haptic feedback from the environment. Later, Goertz was able to build the first electrical teleoperation system, in order to overcome the distance limitations of the mechanically connected systems.
From thereon, teleoperation received an increase in attention and found its usefulness in various applications ranging from space [167, 168] and underwater explorations [169, 170], to nuclear waste decommissioning [171, 172] and surgical robotics [173–175]. The OceanOneK humanoid robot developed by Stanford [176] is controlled via teleoperation in order to perform deep sea exploration of wreck sites, while providing the human in the control room visual and haptic feedback for a heightened sense of presence. In 2001, a surgical procedure was performed with the remote site where the patient was in Strasbourg, France, with the surgeon in New York, USA [177]. Besides providing means for human remote control in inaccessible environments, teleoperation helps overcome the limitations of autonomous robots, which are still not capable of operating completely independently in unstructured environments. Additionally, safety-critical applications such as in the medical field pose ethical and legal concerns in relation to patient safety and accountability, when it comes to fully autonomous robot operation [178]. In such situations, teleoperation allows to combine the human intelligence and high-level decision making, with the accuracy and repeatability of robotic manipulators.
Teleoperating manipulators however can be a challenging task for a number of reasons. To begin with, it takes time and practice to judge distances when looking at the remote manipulator from a computer screen. Moreover, one needs to get a feel of the transformations and scalings needed to map the motion of the master robot to the remote one. That said, teleoperation becomes even more complicated when it is required to control both the remote robot position and orientations, due to the difficulties humans find in rotating objects in 3D [179]. Usually, humans attempt to perform a rotation about only one axis at the time when commanding orientations, in contrast to commanding translations. Furthermore, while force feedback from the remote environment provides the human an immersive experience of the interaction,

contact stability problems are prone to occur, especially in contact with stiff environments. In fact, transparency [180], which defines the range of impedances the master robot can render to the human, often conflicts with stability and a suitable trade-off needs to be found between the two objectives.

Shared Control

These challenges have inspired paradigms where the control of the system is *shared* between the human and the autonomy. The extent to which the autonomy assists the human is flexible and highly depends on the given task and scenario. In [181], the authors classify the different autonomy levels, which range from manual human control, to fully autonomous operation where the robot decides everything and executes the task, ignoring the human. Between the two ends of the spectrum, a continuum of interaction modes exits. For example, the autonomy can offer to the human a set of alternatives to choose from, or execute a particular action only if the human approves it. In the context of telerobotics, control modes can be classified into Direct Control, Supervisory Control and Shared Control [182]. Supervisory Control [183] relies on local autonomy where the robot is capable of performing a set of tasks on its own, while communicating information to the human who provides high-level instructions. This form of control is particularly useful in teleoperation settings where high communication delays are present, such as in space applications. For example, in the METERON project [184], the DLR Rollin' Justin humanoid robot [185] operated in the International Space Station was commanded in supervisory control mode via a tablet interface to perform autonomously tasks such as wiping and plugging cables.
On the other hand, in this thesis, we focus on shared control [186–188], where the interaction between the human and the autonomy is tighter and more dynamic. This can happen in the context of teleoperation, but also in scenarios where the human interacts with the robot in a collocated manner such as in cooperative manipulation tasks. The central idea in shared control is that a human interacts in a continuous manner with an Autonomous Agent (AA) that encodes prior knowledge, in order to achieve a desired task, and with the aim to reduce the mental workload of the human and improve task execution. This however poses several questions with regard to the form and degree of assistance that should be delivered to the human. Clearly, this depends on several factors such as the level of human expertise and confidence, and obviously the task context. For example, in heart surgery, shared control can take the form of compensating for the beating heart movements of the patient to allow for more stable operation [189]. On the other hand, for complex tasks that consist of several subtasks, shared control aims at the smart allocation of these subtasks between the human and the autonomy. This applies for instance in the context of teleoperating a humanoid with a large number of DOF, where the human is given charge of commanding the hand location, while the whole body controller automatically follows this motion and enforces other constraints such as keeping the robot balanced [190]. Similarly, in a dual arm manipulation scenario, the autonomous

controller optimizes the grasping pose and controls the camera location of the two manipulators, in order to facilitate the sorting of nuclear wastes [191].
Over the past few years, several shared control architectures were proposed in literature ranging from partitioned and shared space control, to Virtual Fixtures and haptic shared control. In the partitioned-space architecture, the autonomy controls a subset of the degrees of freedom, while the human is in charge of the rest. For example, the human controls the robot translational position, while the autonomy commands the orientation [192]. On the other hand, in the shared space architecture, the human and the AA command the same DOF, where the human input is fused with the outputs of the autonomous agent depending on some authority allocation metric [193, 194] that determine the weighting between the two commands.
Another widely popular approach in shared control are the so-called Virtual Fixtures (VF) introduced in [195], and which refer to the software-generated positions or forces applied to guide the human. To best illustrate the idea of VF, the common metaphorical example used is that they act like "a ruler guiding a pencil to draw a straight line". Two types of VF can be distinguished in this regard [196], namely, Guidance (GVF) and Forbidden Region Virtual Fixtures (FRVF). In GVF, the goal is to guide the human to a certain goal location or along desired paths or surfaces. On the other hand, in FRVF, the user is prevented from entering into forbidden regions of the workspace, for example to avoid obstacles. In [195], the authors employ FRVF to assist the user in a peg-in-hole assembly task. Depending on the available hardware interface, VF can be of admittance or impedance types, outputting a command velocity or force respectively. The latter types can be seen as a form of Haptic Shared Control, which defines an architecture where both the human and the AA impose forces on the master interface, the output of which is commanded to the remote manipulator.

LfD in Shared Control: The manual design of shared control algorithms can be a daunting task that needs expert knowledge, and would require re-programming to transfer the approach to different contexts. Recently, LfD has been introduced for the design of shared control techniques. Thereby, task knowledge is encoded in a regression model based on demonstrations typically provided by an expert, which can subsequently be used to guide a novice user achieve the desired task. This can be the case for example in surgical procedures to help train a novice surgeon perform certain surgical maneuvers [197]. In [198], the authors integrated LfD into the Shared Control Templates framework [199] to learn constraints and mappings that refine the user input throughout different task stages for teleoperating the assistive robot EDAN [200]. In a series of studies [192, 201, 202], the authors explore the use task-parameterized learning, incremental learning and model predictive control for a partitioned-space teleoperation architecture, in order to perform an ROV task, while overcoming communication delays and possible workspace differences between the master and remote robot. A Shared-Space Control architecture was compared to Haptic Shared Control in [203] for a teleoperated protection cover replacement task, and where LfD in the form of GMM was used. GMM were also deployed in [204] and [205] to design VF that guide the user to one of the possible goal locations,

depending on the probability of each. Along the same lines, in [206], probabilistic movement primitives were combined with the flow controller from [207] for guidance generation. In [208], the authors suggest Locally Weighted Regression (LWR) to encode human demonstrations in order to provide a time-indexed trajectory for an impedance controller that provides a guiding force, with a spring stiffness inversely proportional to the variance in demonstrations. They also propose incremental learning for refining the desired motions. On the other hand, the work in [209] exploits DMP to predict the evolution of one translational DOF, depending on the human state, who controls the other DOF. Incremental learning is also used to refine task knowledge due to a change in the environment. In [210], for the teleoperation of hydraulic manipulators, LfD-based VF are devised to enable path-tracking and obstacle avoidance.

Shared Control For Contact Tasks: While the majority of shared control literature has primarily focused on research problems and applications that mainly concern free motion, fewer works considered tackling tasks that involve continuous physical interaction with the environment. This is typically the case in the surgical domain where tasks such as suturing, cutting, and knot tying essentially involve a robot manipulating a tool in contact with the remote environment. For example, in [211], the authors proposed the use of vector fields to implement forbidden-plane and angle ellipsoid constraints to boost the safety and performance of robots performing a bi-manual cutting task. In [212], the authors propose looping VF and guidance cylinders based on constrained optimization for assistance during a suturing task. The authors of [213] considered robotic cutting, and devised VF that impose non-holonomic constraints on the operator to avoid lateral movements and rotations of the cutting scalpel. VF were also deployed in [214], where haptic cues improve the surgeon's ability to properly guide and constrain the remote robot motion during needle passing and knot tying tasks. In [215], the authors adapt the nominal dissection path based on vision, while in [64] the path is obtained by fitting a polynomial to the via-points provided by the surgeon. In both cases, a VF is used on the master device for haptic guidance. Additionally, in [64], the VF is complemented with an energy tank to ensure passivity, and an impedance shaping function that tunes the strength of the fixutre. Recently, the authors in [216] employed LfD for intent recognition, and subsequently devised VF to assist the human in a collaborative drilling task, using an admittance controlled robot.

In the context of learning, the authors in [217] proposed the use of LfD for a guided peg-in-hole task, where the sensed external forces dictate the guidance trajectory for optimal peg alignment. The authors in [218] propose the concept of corrective shared autonomy for the interactive learning and refinement of contact tasks in the context of aircraft manufacturing, by relying on operator corrections provided via a joystick device.

2.2 Technical Preliminaries

In the remaining part of this chapter, we provide a basic explanation of the core theoretical concepts used in this thesis. We begin by discussing stability and passivity theories, followed by basic robot modeling and control techniques. Finally, we offer a concise overview on some machine learning regression methods.

2.2.1 Stability Theory

In this section, we briefly review some of the important concepts related to stability theory [219, 220]. For the following definitions, we consider a state $\boldsymbol{\xi} \in \mathbb{R}^d$ with dimension d. This can be used to describe the position and/or the velocity of a robot. The evolution of the state is governed according to a dynamical system given by

$$\dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{\xi}) \tag{2.1}$$

where $\boldsymbol{f}() : \mathbb{R}^d \mapsto \mathbb{R}^d$ is a continuously differentiable function that maps the current state into the rate of change of the state with respect to time. Although (2.1) does not explicitly feature a control input $\boldsymbol{u}(\boldsymbol{\xi})$, it should be noted that (2.1) is quite flexible in modeling a feedback controlled system. Clearly, writing $\dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{\xi}, \boldsymbol{u}(\boldsymbol{\xi}))$ is still captured in the representation (2.1), and therefore (2.1) can be adequately used to describe the open/closed-loop dynamics of a system. A solution $\boldsymbol{\xi}(t)$ of (2.1) describes a state trajectory, which refers to the evolution of the state from the initial condition. Additionally, the dynamical system (2.1) is said to be time-invariant or autonomous since it does not explicitly depend on time, while a system with the dynamics $\dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{\xi}, t)$ is said to be non-autonomous.
In the following, to analyze the stability properties of (2.1), it becomes useful to define a sphere $\boldsymbol{S}_R$ with radius R and a ball region $\boldsymbol{B}_R$ defined by $||\boldsymbol{\xi}|| < R$. Without loss of generality, we assume that the equilibrium of the system is at the origin i.e $\boldsymbol{\xi} = \mathbf{0}$.

Definition 1 *The equilbruim state $\boldsymbol{\xi} = \mathbf{0}$ is said to be stable if there exists $r > 0$ such that if $||\boldsymbol{\xi}(0)|| < r$, then $||\boldsymbol{\xi}(t)|| < R$ for all $t \geq 0$*

which essentially means that a system trajectory remains arbitrarily close to the equilibrium if it starts close enough to it. If for some r the initial state is within $\boldsymbol{B}_r$, then all state trajectories will remain in $\boldsymbol{B}_r$. If in addition, the state converges exactly to the equilibrium, then the equilibrium point is asymptotically stable.

Definition 2 *The equilbruim state $\boldsymbol{\xi} = \mathbf{0}$ is said to be asymptotically stable if for some $r > 0$ and $||\boldsymbol{\xi}(0)|| < r$, $\lim_{t\to\infty} \boldsymbol{\xi}(t) = \mathbf{0}$*

and where $\boldsymbol{B}_r$ is the *domain of attraction* i.e all trajectories starting within the ball $\boldsymbol{B}_r$ eventually converge to the equilibrium. If $\boldsymbol{B}_r$ covers the entire state space, such that for any initial state, we have $\lim_{t\to\infty} \boldsymbol{\xi}(t) = \mathbf{0}$, then the equilibrium point is said to be globally asymptotically stable.
To analyze stability of dynamical systems, a powerful tool that can be effectively

employed is Lyapunov direct method, which exploits a scalar function of the state $\boldsymbol{V}(\boldsymbol{\xi}) : \mathbb{R}^d \mapsto \mathbb{R}$, analogous to the energy of a mechanical system. The basic intuition is that if for a mechanical system, its total energy is continuously dissipated until the system is at rest at the equilibrium, that we can conclude the stability the system. This can be formalized as

Theorem 1 *For the system* (2.1)*, if within the ball* $\boldsymbol{B}_r$ *there exists a scalar function* $\boldsymbol{V}(\boldsymbol{\xi})$ *such that*

- $V(\mathbf{0}) = 0$ *and* $V(\boldsymbol{\xi}) > 0 \; \forall \boldsymbol{\xi} \neq 0$
- $\dot{V}(\boldsymbol{\xi}) = \frac{\partial V(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \boldsymbol{f}(\boldsymbol{\xi}) \leq 0$

then the equilibrium is stable, and is locally asymptotically stable if $\dot{V}(\boldsymbol{\xi}) < 0 \, \forall \, \boldsymbol{\xi} \neq 0$*. Furthermore, if* $V(\boldsymbol{\xi})$ *is radially unbounded, such that* $||\boldsymbol{\xi}|| \to \infty \implies V(\boldsymbol{\xi}) \to \infty$*, then the equilibrium is globally asymptotically stable.*

It happens quite often that for an asymptotically stable system, we have $\dot{V}(\boldsymbol{\xi}) \leq 0$ which allows us to conclude only stability but not asymptotic convergence. This is the case for example for a mass-spring-damper system with constant equilibrium position. In such situations, if the system is autonomous, then LaSalle invariance principle offers a powerful solution to establish asymptotic stability even for $\dot{V}(\boldsymbol{\xi}) \leq 0$. The principle relies on the idea of invariance of a set $\boldsymbol{G}$, which means that any trajectory starting in $\boldsymbol{G}$ remains in $\boldsymbol{G}$ for all future times. We can then state the following:

Theorem 2 *For the system* (2.1)*, if there exists scalar function* $V(\boldsymbol{\xi})$ *with continuous first partial derivatives, such that*

- $||\boldsymbol{\xi}|| \to \infty \implies V(\boldsymbol{\xi}) \to \infty$
- $\dot{V}(\boldsymbol{\xi}) \leq 0$
- E *is the set of all points for which* $\dot{V}(\boldsymbol{\xi}) = 0$ *i.e* $E = \{\boldsymbol{\xi} \in \mathbb{R}^d | \dot{V}(\boldsymbol{\xi}) = 0\}$
- $\boldsymbol{G}$ *is the largest invariant set in* $\boldsymbol{E}$

Then all solutions globally asymptotically converge to $\boldsymbol{G}$*. Furthermore if* $\boldsymbol{G}$ *contains no solutions other than the the trivial solution* $\boldsymbol{\xi} = \mathbf{0}$*, then the origin is globally asymptotically stable.*

2.2.2 Passivity Theory

Closely related to stability, passivity [112] is an important tool that can be used to reason about the stability proprieties of an open/closed-loop system, and has been central to control design in the robotics community, especially in the context of physical interactions [221]. In order to analyze the passivity properties of the system, we consider an affine dynamical system of the form:

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= \boldsymbol{f}(\boldsymbol{\xi}) + \boldsymbol{g}(\boldsymbol{\xi})\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{\xi}) \end{aligned} \tag{2.2}$$

with the smooth vector fields $\boldsymbol{f}(.)$, $\boldsymbol{g}(.)$ and $\boldsymbol{h}(.)$, and with $\boldsymbol{u} \in \mathcal{U}$ $\boldsymbol{y} \in \mathcal{Y}$ being the input and output, respectively, which belong to the m dimensional spaces $\mathcal{U}$ and $\mathcal{Y}$. Let w be a real valued map such that $w : \mathcal{U} \times \mathcal{Y} \to \mathbb{R}$, we call w the supply rate. Let also $S(\boldsymbol{\xi})$ be a continuous positive definite function, called the *storage function*, we can make the following definition [112]

Definition 3 *The system in* (2.2) *is called dissipative, if for any initial condition* $\boldsymbol{\xi}(t_0)$ *at a time* t_0*, for any* $\boldsymbol{u} \in \mathcal{U}$ *and for any* $t_1 > t_0$*, the following inequality holds*

$$S(\boldsymbol{\xi}(t_1)) \leq S(\boldsymbol{\xi}(t_0)) + \int_{t_0}^{t_1} s(u(t), y(t))dt \tag{2.3}$$

and which can be alternatively stated as

$$\dot{S}(\boldsymbol{\xi}(t)) \leq s(u(t), y(t)) \tag{2.4}$$

For an important class of systems, including the systems considered in this thesis, the storage function coincides with the energy of the system, while the supply rate coincides with power, such that $w(\boldsymbol{u}(t), \boldsymbol{y}(t)) = \boldsymbol{y}^T\boldsymbol{u}$ describes the power flow into or out of the system. This leads us to the definition of a passive system:

Definition 4 *The system in* (2.2) *is said to be passive if it is dissipative with the supply rate* $w(\boldsymbol{u}(t), \boldsymbol{y}(t)) = \boldsymbol{y}^T\boldsymbol{u}$

The above definitions essentially mean that for a system to be passive, the storage function (i.e stored energy) at any given time can be at most equal to the initially available storage, in addition to the externally supplied power. In other words, there can be no internal generation of energy, instead, the energy is either stored, or dissipated.
When talking about passivity, we are referring to passivity with respect to an input-output pair i.e $(\boldsymbol{u}, \boldsymbol{y})$. In the context of physical systems, this pair describes a *power port* $P = \mathcal{Y} \times \mathcal{U}$, which represents the entity by means of which the system can be interconnected to, and therefore, exchange energy, with other systems [221]. This port is described by a pair of power-conjugated variables i.e an effort-flow pair whose product describes the power traversing the port. Physical systems are often interconnected by power-preserving interconnections, which means that along the interconnection, energy cannot be lost or generated, but only transferred from one system to another. This idea can be formalized by a powerful mathematical tool called the *Dirac structure*; a geometric structure that represents the energetic topology in the system, interconnecting the different components that exchange energy through their interaction ports. Furthermore, a Dirac structure is power-conserving in the sense the power lost along the structure is zero. Let $\mathcal{F}$ be the linear vector space with the corresponding flow variables $f \in \mathcal{F}$, its dual subspace $\mathcal{E} = \mathcal{F}^*$ that represents the effort variables $e \in \mathcal{E}$, a Dirac structure can be defined as [221]

Definition 5 *A Dirac structure on the vector space* $\mathcal{E} * \mathcal{F}$ *is a subspace* $\mathcal{D}$ *such that* $< e, f >= 0 \ \forall (e, f) \in \mathcal{D}$

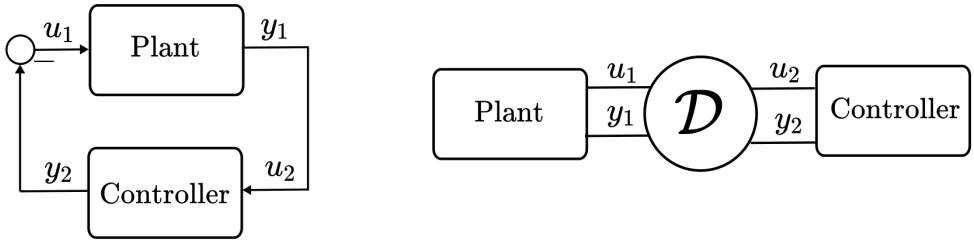


Fig. 2.1: Example of a Dirac structure, presented in the form of a feedback interconnection

where $< e, f >= e^T f$ is the duality product. Theorem 1 implies power conservation which means that the power entering the structure is equal to the power leaving it, i.e the net power entering or leaving is zero. A simple example of a Dirac structure is the feedback interconnection between a plant and a controller (Fig. 2.1). The interconnection

$$\begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & -I \\ I & \mathbf{0} \end{bmatrix}}_{\mathcal{D}} \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} \tag{2.5}$$

is power-preserving since $\boldsymbol{u}_1^T \boldsymbol{y}_1 = -\boldsymbol{u}_2^T \boldsymbol{y}_2$ indicating that power is exchanged between the plant and the controller without any losses.

Energy Tanks

Energy tanks [149], [222] have been often used to guarantee the passivity of a controlled system. An energy tank is an atomic energy storage element (i.e a one dimensional spring), initialized by an energy budget allocated to passively implement any control action that can potentially violate the passivity of the system. This budget can be set empirically, or by a high-level supervisor, that takes into account task-related aspects or certain safety metrics. Formally, the tank can be modeled as a dynamical system, with a state $x_t \in \mathbb{R}$, an input $u_t \in \mathbb{R}$, an output $y_t \in \mathbb{R}$, an energy function $E_t = \frac{1}{2} x_t^2$, and an initial energy budget E_0. In the simplest case[2], the dynamics of the tank take the following form

$$\dot{x}_t = u_t \tag{2.6}$$

$$y_t = x_t \tag{2.7}$$

which clearly constitutes a passive system with respect to the pair (u_t, y_t). For ease of exposition, let us consider a simplified point mass robot $\ddot{x} = u_c' + u_e$, with an output $\dot{x}$, an arbitrary non-passive control input u_c' and an interaction force with the environment u_e. To compensate for u_c', the tank is interconnected to the system in a power-preserving manner via a Dirac structure, modulated by u_c'. This can be

[2] If the original control input includes damping, then the dissipated energy can optionally be used to replenish the tank

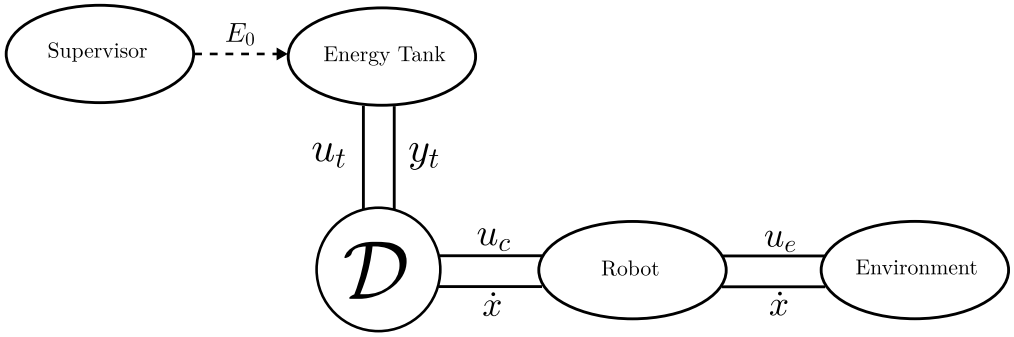


Fig. 2.2: Port-Based Model of different subsystems interconnected together. The energy tank is interconnected via a Dirac structure $\mathcal{D}$ to the robot, which interacts with the environment. The supervisor represents an abstract entity that takes high-level decisions regarding the initial energy budget, as well as when and how to replenish the tank.

represented as

$$\begin{bmatrix} u_t \\ u_c \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -\frac{w}{x_t} \\ \frac{w}{x_t} & 0 \end{bmatrix}}_{\mathcal{D}} \begin{bmatrix} x_t \\ \dot{x} \end{bmatrix} \tag{2.8}$$

where the $w = \sigma u_c'$, with σ given by

$$\sigma = \begin{cases} 1, & \text{if } E_t \geq \underline{E}_t \,, \\ 0, & \text{otherwise} \end{cases} \tag{2.9}$$

where $\underline{E}_t > 0$ is a user-defined lower limit for the energy in the tank, while u_c is the passive implementation of u_c' now commanded to the robot, such that $\ddot{x} = u_c + u_e$. Intuitively, the role of σ can be seen as a valve that allows energy flow from the tank into the system as long as the energy in the tank is not depleted, and therefore, guaranteeing that the energy the controller injects into the system remains bounded. Clearly, the structure $\mathcal{D}$ is power-preserving due to its skew-symmetry. The overall passivity of the system can be easily shown with the storage $S_t = \frac{1}{2}\dot{x}^2 + E_t$, resulting in the energy balance $\dot{S}_t = \dot{x}u_e$, proving the passivity of the system with respect to the port $(\dot{x}, u_e)$, and which in consequence proves the stable interaction of the system with a passive environment [149]. Fig. 2.2 shows a conceptual illustration of a system passified by an energy tank, represented as several subsystems exchanging energy.

2.2.3 Robot Modeling

Throughout the thesis, we consider robots with the following rigid body dynamics

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{ext}} \,, \tag{2.10}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ is the joint position, while n is dimension of the joint space. $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is the symmetric positive-definite inertia matrix, and $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n \times n}$ is the Coriolis and centrifugal matrix, which satisfies $\dot{\boldsymbol{M}}(\boldsymbol{q}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{C}^T(\boldsymbol{q}, \dot{\boldsymbol{q}})$. The

gravity torque is given by $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$, while the control and external torques are denoted by $\boldsymbol{\tau} \in \mathbb{R}^n$ and $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^n$, respectively.
The Cartesian end-effector coordinates are denoted by $\boldsymbol{x} \in \mathbb{R}^m$ where m is the number of Cartesian DOF, e.g. where $m = 6$ for a complete characterization of the end-effector positions and orientations. It is possible to relate the joint positions to the cartesian positions through the forward kinematics map $\boldsymbol{f}_k(.)$, such that $\boldsymbol{x} = \boldsymbol{f}_k(\boldsymbol{q})$. On a differential level, the mapping between the joint-space velocities to the task-space velocities is obtained via $\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$, where $\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{f}_k(\boldsymbol{q})}{\partial \boldsymbol{q}}$ is the end-effector Jacobian. Additionally, we can also relate the Cartesian forces $\boldsymbol{F} \in \mathbb{R}^m$ to the corresponding joint torques through the relation $\boldsymbol{\tau} = \boldsymbol{J}^T\boldsymbol{F}$. This relation becomes very useful to command actuator torques from task-space control forces, without the inversion of the task Jacobian.
It is quite common to express the control goal for a robot in the task-space. Therefore, it is convenient to also define the operational space dynamics. For the non-redundnant case where $m = n$, this can be expressed as [38]

$$\boldsymbol{\Lambda}_x(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{\mu}_x(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{F}_g(\boldsymbol{x}) = \boldsymbol{F} + \boldsymbol{F}_{ext}\,, \tag{2.11}$$

where $\boldsymbol{\Lambda}_x = \boldsymbol{J}^{-T}\boldsymbol{M}\boldsymbol{J}^{-1}$ is the task-space inertia, $\boldsymbol{\mu}_x = \boldsymbol{\Lambda}_x(\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{C} - \dot{\boldsymbol{J}})\boldsymbol{J}^{-1}$ is the task-space Coriolis and centrifugal matrix, and $\boldsymbol{F}_g(\boldsymbol{x})$ are the task-space gravity forces. The Cartesian control and external forces acting on the end-effector are denoted $\boldsymbol{F}$ and $\boldsymbol{F}_{ext}$, respectively.
The dynamics in (2.10) and (2.11) satisfy the following properties:

- The matrices $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{\Lambda}_x(\boldsymbol{x})$ are symmetric positive definite matrices.
- The matrices $\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\dot{\boldsymbol{\Lambda}}_x(\boldsymbol{x}) - 2\boldsymbol{\mu}_x(\boldsymbol{x}, \dot{\boldsymbol{x}})$ are skew-symmetric.

The important role these properties play becomes apparent when analyzing the passivity of the dynamics (2.10), with the following storage function:

$$S(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{V}_g(\boldsymbol{q}) \tag{2.12}$$

where $\boldsymbol{V}_g(\boldsymbol{q})$ is a conservative potential from which the gravity torque field is generated, such that $\boldsymbol{g}(\boldsymbol{q}) = -\frac{\partial V_g(\boldsymbol{q})}{\partial \boldsymbol{q}}$. Note that thanks to the first property, we are able to use $\boldsymbol{M}(\boldsymbol{q})$ as part of the storage function in order to describe the kinetic energy of the robot, while the second term in S is the potential energy due to gravity. Taking the derivative of S with respect to time, we have

$$\dot{S} = \frac{1}{2}\dot{\boldsymbol{q}}^T(\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}))\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T(\boldsymbol{\tau} + \boldsymbol{\tau}_{\text{ext}}) \tag{2.13}$$

where thanks to the skew-symmetry property, we have $\dot{S} = \dot{\boldsymbol{q}}^T(\boldsymbol{\tau} + \boldsymbol{\tau}_{\text{ext}})$ which means that the robot dynamics are passive with respect to the port $(\dot{\boldsymbol{q}}, \boldsymbol{\tau} + \boldsymbol{\tau}_{ext})$. An important consequence of this passivity feature is that designing passive controllers with respect to the pair $(\dot{\boldsymbol{q}}, -\boldsymbol{\tau})$ results in the overall passivity of the robot with respect to its port of interaction with the environment $(\dot{\boldsymbol{q}}, \boldsymbol{\tau}_{ext})$.

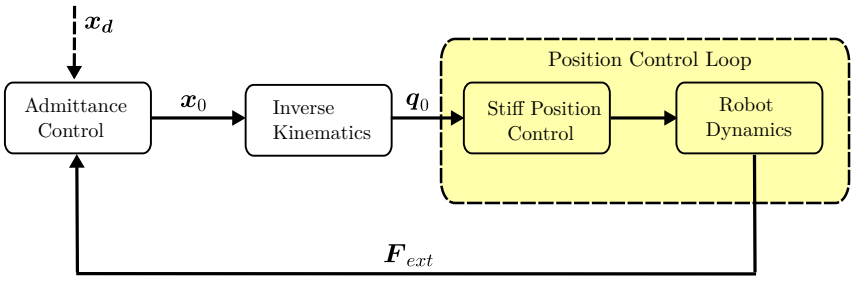


Fig. 2.3: Block diagram of an admittance-controlled robot with a stiff position control inner loop. The virtual desired equilibrium position is denoted $\boldsymbol{x}_d$, while $\boldsymbol{x}_0$ is the actual desired position obtained by simulating the admittance dynamics.

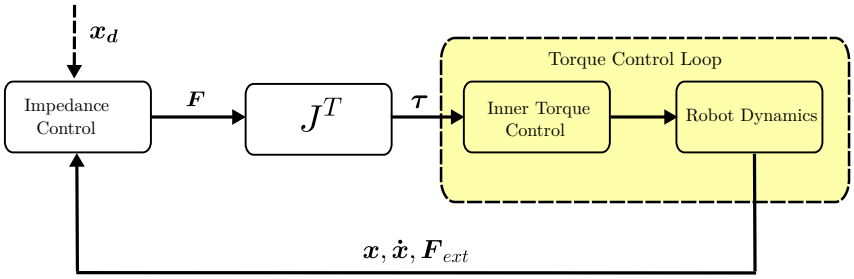


Fig. 2.4: Block diagram of an impedance-controlled robot with a torque control inner loop. Similar to Fig. 2.3, $\boldsymbol{x}_d$ refers to the desired equilibrium.

2.2.4 Impedance Control

As stated earlier, the goal of impedance control [4] is to establish a dynamic relationship between the manipulator and the environment, at the ports of interaction i.e where the manipulator physically interacts with the environment e.g. at the end-effector. Following the postulate that a controlled manipulator behaves fundamentally as a physical system, we can model the manipulation problem as two physical systems interacting together. A physical system can be either modeled as an admittance which accepts an effort (e.g. a force) and yields a flow (e.g. motion), or an impedance that accepts a flow and outputs an effort in response. In the original paper [4], N. Hogan argued that since most of the environments a robot interacts with come in the form of an admittance e.g. inertial masses, the controlled manipulator should admit an impedance causality for physical compatibility.
The concept of impedance control is widely used nowadays in several robotic domains. In the literature, two popular control implementations can be distinguished and are often used to achieve the same goal of the original impedance control framework: impedance control and admittance control [45]. Impedance control often refers to the case where the controller itself has an impedance causality and outputs a control force. This type of implementation is commonly used in lightweight robots that provide a low-level torque control interface, such as the Kuka LWR and Panda Franka Emika. On the other hand, in admittance control, the desired dynamic behavior is simulated producing a desired motion in response to a force, which is then fed to (stiff) inner

position-control loop. This implementation strategy is very suitable for industrial robots that are often position-controlled. It has been noted that admittance control can lead to stability problems if the inner loop is not stiff enough, or for the feedback of non-collocated forces [223]. Figs. 2.3 and 2.4 show block diagrams for admittance and impedance control, respectively. In this thesis, impedance control refers to the first control implementation.

Cartesian Impedance Control

For most of the tasks considered in this thesis and generally in manipulator control, the physical interaction with the environment happens at the end-effector. Given a desired twice differentiable equilibrium trajectory $\boldsymbol{x}_d(t)$, the goal of impedance control in end-effector coordinates can be expressed as

$$\boldsymbol{\Lambda}_d\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d\tilde{\boldsymbol{x}} = \boldsymbol{F}_{ext}\,, \tag{2.14}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}$ is the task space error, while $\boldsymbol{\Lambda}_d \in \mathbb{R}^{m\times m}$, $\boldsymbol{D}_d \in \mathbb{R}^{m\times m}$ and $\boldsymbol{K}_d \in \mathbb{R}^{m\times m}$ are respectively the desired task space intertia, damping and stiffness. These matrices can be kept constant, or allowed to vary in time or state depending on the task requirements. In order for (2.14) to be physically meaningful, it is necessary however that these matrices stay positive definite, which also ensures that the closed-loop dynamics are stable.
One possible way to obtain the target behavior (2.14) can be realized with the following control law

$$\boldsymbol{\tau} = \boldsymbol{J}^T\boldsymbol{F} \tag{2.15}$$

$$\begin{aligned}\boldsymbol{F} =& \boldsymbol{F}_g(\boldsymbol{x}) + \boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}}_d + \boldsymbol{\mu}(\boldsymbol{x},\dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \\ & \boldsymbol{\Lambda}(\boldsymbol{x})\boldsymbol{\Lambda}_d^{-1}(\boldsymbol{K}_d\tilde{\boldsymbol{x}} + \boldsymbol{D}_d\dot{\tilde{\boldsymbol{x}}}) + (\boldsymbol{\Lambda}(\boldsymbol{x})\boldsymbol{\Lambda}_d^{-1} - \boldsymbol{I})\boldsymbol{F}_{ext},\end{aligned} \tag{2.16}$$

which corresponds to the PD+ control law [33]. The stability properties of this controller can be analyzed with the time-varying Lyapunov function $V = \frac{1}{2}\dot{\tilde{\boldsymbol{x}}}^T\boldsymbol{\Lambda}(\boldsymbol{x})\dot{\tilde{\boldsymbol{x}}} + \frac{1}{2}\tilde{\boldsymbol{x}}^T\boldsymbol{K}_d\tilde{\boldsymbol{x}}$, and where for constant positive-definite impedance matrices, the asymptotic convergence of the tracking errors $\tilde{\boldsymbol{x}} \to \boldsymbol{0}$ and $\dot{\tilde{\boldsymbol{x}}} \to \boldsymbol{0}$ can be proved.
Note however that control law (2.15) requires the feedback of the external forces $\boldsymbol{F}_{ext}$ which is often not desirable. This a consequence of the requirement that a desired inertial behavior is specified. To avoid this problem, it is common to avoid inertia shaping by setting $\boldsymbol{\Lambda}_d = \boldsymbol{\Lambda}(\boldsymbol{x})$, and therefore feedback of $\boldsymbol{F}_{ext}$ can be avoided[3].
For the regulation case, where $\dot{\boldsymbol{x}}_d = \boldsymbol{0}$, a popular control law to realize the desired impedance behavior without inertia shaping is the famous PD+gravity compensation controller [31], expressed as

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{K}_d\tilde{\boldsymbol{x}} - \boldsymbol{D}_d\dot{\boldsymbol{x}}) + \boldsymbol{g}(\boldsymbol{q}), \tag{2.17}$$

which results in the following closed-loop dynamics

$$\boldsymbol{\Lambda}_x(\boldsymbol{x})\ddot{\boldsymbol{x}} + (\boldsymbol{D}_d + \boldsymbol{\mu}_x(\boldsymbol{x},\dot{\boldsymbol{x}}))\dot{\boldsymbol{x}} + \boldsymbol{K}_d\tilde{\boldsymbol{x}} = \boldsymbol{F}_{ext}\,. \tag{2.18}$$

[3] Impedance control without inertia shaping is often referred to as 'compliance control'

While the above dynamics still feature the Coriolis and centrifugal term $\boldsymbol{\mu}_x(\boldsymbol{x}, \dot{\boldsymbol{x}})$, for relatively slow robot motions, this term is negligible and therefore, the dynamics (2.18) is practically equivalent to the target dynamics (2.14). In addition to its simple implementation, a particularly appealing aspect of (2.17) are the nice stability properties this controller possess. Consider the following Lyapunov function:

$$V(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\tilde{\boldsymbol{x}}^T \boldsymbol{K}_d \tilde{\boldsymbol{x}}, \tag{2.19}$$

with rate along the closed-loop dynamics (2.18) given by

$$\dot{V} = \dot{\boldsymbol{x}}^T \boldsymbol{F}_{ext} - \dot{\boldsymbol{x}}^T \boldsymbol{D}_d \dot{\boldsymbol{x}}. \tag{2.20}$$

For the case where robot is in free motion i.e $\boldsymbol{F}_{ext} = \boldsymbol{0}$, we have $\dot{V} = \dot{\boldsymbol{x}}^T \boldsymbol{D}_d \dot{\boldsymbol{x}} \leq 0$ and therefore the system is stable. By invoking LaSalle invariance principle, it is possible to conclude the asymptotic stability of the equilibrium $(\boldsymbol{x}_d, \boldsymbol{0})$. Additionally, during physical interactions where $\boldsymbol{F}_{ext} \neq \boldsymbol{0}$, we have $\dot{V} \leq \dot{\boldsymbol{x}}^T \boldsymbol{F}_{ext}$ which means that the controlled robot is passive with respect to port $(\dot{\boldsymbol{x}}, \boldsymbol{F}_{ext})$ through which the robot interacts (i.e exchanges energy) with the environment, and in consequence, the stable interaction with arbitrary passive environments is guaranteed [149].

Joint Impedance Control

In some cases, it is desired to specify the impedance behavior on joint-level. This becomes very relevant in redundancy resolution schemes, where it is common to specify the secondary task as a joint-level impedance/compliance. Given a desired smooth equilibrium trajectory $\boldsymbol{q}_d(t)$, and $\boldsymbol{M}_d \in \mathbb{R}^{n\times n}$, $\boldsymbol{D}_d \in \mathbb{R}^{n\times n}$, $\boldsymbol{K}_d \in \mathbb{R}^{n\times n}$ as symmetric positive definite impedance parameters, the nominal interaction behavior can be specified as

$$\boldsymbol{M}_d \ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{D}_d \dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_d \tilde{\boldsymbol{q}} = \boldsymbol{\tau}_{ext}\,, \tag{2.21}$$

where $\tilde{\boldsymbol{q}} = \boldsymbol{q}_d(t) - \boldsymbol{q}$ is the joint space error. Similar to the Cartesian case, it is possible to realize the prescribed behavior through different control implementations. For instance, it is possible to adapt the controller from (2.15) to the joint case, resulting in the following controller

$$\begin{aligned}\boldsymbol{\tau} =& \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_d + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \\ & \boldsymbol{M}(\boldsymbol{q})\boldsymbol{M}_d^{-1}(\boldsymbol{K}_d\tilde{\boldsymbol{q}} + \boldsymbol{D}_d\dot{\tilde{\boldsymbol{q}}}) + (\boldsymbol{M}(\boldsymbol{q})\boldsymbol{M}_d^{-1} - \boldsymbol{I})\boldsymbol{\tau}_{ext},\end{aligned} \tag{2.22}$$

which similar to before, require the feedback of external torques $\boldsymbol{\tau}_{ext}$ acting on the robot. In case inertia shaping is avoided i.e $\boldsymbol{M}_d = \boldsymbol{M}(\boldsymbol{q})$, and for the regulation case where the $\boldsymbol{q}_d$ is constant, the PD+gravity compensation control law can be also formulated in joint space as

$$\boldsymbol{\tau} = \boldsymbol{K}_d\tilde{\boldsymbol{q}} - \boldsymbol{D}_d\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \tag{2.23}$$

and where analogously to the Cartesian case, we can prove passivity with respect to the port $(\dot{\boldsymbol{q}}, \boldsymbol{\tau}_{ext})$ and asymptotic convergence to the equilibrium $\boldsymbol{q}_d$.

2.2.5 Regression Approaches

In this section, we briefly describe the main regression concepts used in thesis. Throughout this section, the multidimensional vectors $\boldsymbol{\Psi}_I \in \mathbb{R}^{N \times d_I}$ and $\boldsymbol{\Psi}_O \in \mathbb{R}^{N \times d_O}$ denote the input and output data matrices respectively, which consist of N data point vectors $\boldsymbol{\xi}_I \in \mathbb{R}^{d_I}$ and $\boldsymbol{\xi}_O \in \mathbb{R}^{d_O}$, while d_I and d_O refer to the input and output dimensions. In robotic applications, such data can be obtained via demonstrations provided by the human through modalities such as kinesthetic teaching or teleoperation [77]. The goal is to to find a suitable model that can accurately map the input features to their correspond output.

Linear Regression

In linear regression, our aim is to find a linear model between the input and output data, through a matrix $\boldsymbol{B} \in \mathbb{R}^{d_I \times d_O}$, such that $\boldsymbol{\Psi}_O = \boldsymbol{\Psi}_I \boldsymbol{B}$. This can be obtained by solving the following least squares problem

$$\min_{\boldsymbol{B}} \quad \sum_{t=0}^{T} ||\boldsymbol{\Psi}_o - \boldsymbol{\Psi}_I \boldsymbol{B}|| \tag{2.24a}$$

which results in the well-known pseudo-inverse solution $\boldsymbol{B} = \boldsymbol{\Psi}_I^T (\boldsymbol{\Psi}_I \boldsymbol{\Psi}_I^T)^{-1} \boldsymbol{\Psi}_O$.
In order to improve regression performance, it is common to include a regularization term to the cost function, which helps improve generalization and prevent overfitting.

Locally Weighted Regression

The main limitation in Linear Regression is that it assumes a linear relationship between input and output variables. To solve this, Locally Weighted Regression can be a suitable alternative to capture more complex relationships, by modeling the data using a set of K local linear models combined together via a weighted sum. The weights can be chosen as radial basis functions, defined as

$$\tilde{w}(\boldsymbol{\psi}_I) = \exp(-\frac{1}{2}(\boldsymbol{\psi}_I - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\psi}_I - \boldsymbol{\mu}_k) \tag{2.25}$$

with centroids $\boldsymbol{\mu}_k$ chosen to be uniform across the input space and band-width $\boldsymbol{\Sigma}_k$. The weights are then normalized as $w(\boldsymbol{\psi}_I) = \frac{\tilde{w}(\boldsymbol{\psi}_I)}{\sum_{k=1}^{K} \tilde{w}(\boldsymbol{\psi}_I)}$. Subsequently, we compute a diagonal matrix $\boldsymbol{W}_k$ that stacks the weights computed over all N points. Finally, the output is related to the input as:

$$\boldsymbol{\Psi}_O = \sum_{k=1}^{K} \boldsymbol{W}_k \boldsymbol{\Psi}_i \boldsymbol{B}_k \tag{2.26}$$

where we have to solve a weighted linear regression problem K times, yielding the solution $\boldsymbol{B}_k = \boldsymbol{W}_k \boldsymbol{\Psi}_I^T (\boldsymbol{\Psi}_I \boldsymbol{W}_k \boldsymbol{\Psi}_I^T)^{-1} \boldsymbol{\Psi}_O$.

Gaussian Mixture Regression

Another powerful tool to model input-output relationships is Gaussian Mixture Models (GMM) [77]. The main idea in GMM is to represent a complex probability distribution as a weighted combination of Gaussians, instead of a single Gaussian which might not be sufficient to capture complex relationships. This technique has found its usefulness in a wide range of applications such speech recognition, medical imaging and robotics. Let $\boldsymbol{\xi} = [\boldsymbol{\xi}_I, \boldsymbol{\xi}_O]$ be a multi-dimensional variable, GMM models the joint probability distribution of $\boldsymbol{\xi}$ using the following density function

$$\begin{aligned} \mathcal{P}(\boldsymbol{\xi}) &= \sum_{n=1}^{N} \pi_n \mathcal{N}(\boldsymbol{\xi}, \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n) \ , \\ \boldsymbol{\mu}_n &= \begin{pmatrix} \boldsymbol{\mu}_n^i \\ \boldsymbol{\mu}_n^o \end{pmatrix} \ , \ \boldsymbol{\Sigma}_n = \begin{bmatrix} \boldsymbol{\Sigma}_n^i & \boldsymbol{\Sigma}_n^{i,o} \\ \boldsymbol{\Sigma}_n^{o,i} & \boldsymbol{\Sigma}_n^o, \end{bmatrix} \end{aligned} \tag{2.27}$$

where N is the number of Gaussian models, which can be set empirically, or determined optimally from the data via e.g the Bayesian-Information-Criterion (BIC). π_n, $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$ are the priors, means and covariances of each Gaussian, respectively. In order to learn these parameters from demonstrations, we can apply for example the well-known iterative Expectation-Maximization (EM) algorithm. During reproduction, first, we need to obtain the distribution of the output conditioned on the input for each time step. This can be achieved via Gaussian Mixture Regression (GMR) as

$$\begin{aligned} \mathcal{P}(\boldsymbol{\xi}_O|\boldsymbol{\xi}_I) &= \sum_{n=1}^{N} h_n(\boldsymbol{\xi}_I) \mathcal{N}(\boldsymbol{\xi}, \hat{\boldsymbol{\mu}}_n^O, \hat{\boldsymbol{\Sigma}}_n^O), \\ \hat{\boldsymbol{\mu}}_n^O &= \boldsymbol{\mu}_n^O + \boldsymbol{\Sigma}_n^{O,I} {\boldsymbol{\Sigma}_n^I}^{-1} (\boldsymbol{\xi}_I - \boldsymbol{\mu}_n^I), \\ \hat{\boldsymbol{\Sigma}}_n^O &= \boldsymbol{\Sigma}_n^O - \boldsymbol{\Sigma}_n^{O,I} {\boldsymbol{\Sigma}_n^I}^{-1} \boldsymbol{\Sigma}_n^{I,O}, \\ h_n(\boldsymbol{\xi}_I) &= \frac{\pi_n \mathcal{N}(\boldsymbol{\xi}_I|\boldsymbol{\mu}_n^I, \boldsymbol{\Sigma}_n^I)}{\sum_{p=1}^{P} \pi_p \mathcal{N}(\boldsymbol{\xi}_I|\boldsymbol{\mu}_p^I, \boldsymbol{\Sigma}_p^I)}. \end{aligned} \tag{2.28}$$

We can then compute the output as the mean of the conditional distribution with

$$\hat{\boldsymbol{\mu}}^O = \sum_{n=1}^{N} h_n(\boldsymbol{\xi}_I) \hat{\boldsymbol{\mu}}_n^O. \tag{2.29}$$

Dynamical Systems

In chapter 6, we make use of first-order DS to encode and generate reference robot motions, due to their nice stability properties and flexibility during learning. Let $\boldsymbol{\xi}$ be a generalized state variable that represents the n dimensional robot joint or cartesian positions, a first-order DS that represents the desired robot motion plan is expressed as

$$\dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{\xi}) \tag{2.30}$$

where $\boldsymbol{f}(.) : \mathbb{R}^d \mapsto \mathbb{R}^d$ is a continuously differentiable function, and $\dot{\boldsymbol{\xi}}$ is the reference velocity. Several approaches have been proposed in the literature to design $\boldsymbol{f}(.)$ e.g learning using GMM or Gaussian processes. In this thesis, we often use the Stable Estimator of Dynamical Systems (SEDS) [13], which is one of the very first formulations proposed for learning $\boldsymbol{f}(.)$. In SEDS, the DS is formulated using GMM as the weighted sum of K linear DS, written as

$$\dot{\boldsymbol{\xi}} = \sum_{k=1}^{K} h_k(\boldsymbol{\xi}) \boldsymbol{A}_k (\boldsymbol{\xi} - \boldsymbol{\xi}^*) \tag{2.31}$$

where $\boldsymbol{\xi}^*$ is the global attractor to which the DS shall converge, while $\boldsymbol{A}_k$ is a constant matrix that describes the dynamics of each linear DS. The non-linear weighting function $h_k(\boldsymbol{\xi})$ is computed as described in the previous section. Learning $\boldsymbol{A}_k$ and the priors π_k in $h_k(\boldsymbol{\xi})$ is formulated as a constrained optimization problem. The cost function is chosen to minimize the discrepancy between $\boldsymbol{f}(\boldsymbol{\xi})$ and human demonstrations consisting of position-velocity pairs. This can be described as the mean-square error or the Log-Likelihood. To guarantee convergence, constraints are added to ensure $\boldsymbol{A}_k + \boldsymbol{A}_k^T \prec 0$ and that all the priors in $h_k(\boldsymbol{\xi})$ are greater than zero with a sum amounting to 1. The asymptotic stability of $\boldsymbol{f}(\boldsymbol{\xi})$ can be formally proven using $V = \frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\xi}^*)^T(\boldsymbol{\xi} - \boldsymbol{\xi}^*)$ as a Lyapunov function.

CHAPTER 3

Hierarchical Passive Variable Impedance Control

This chapter presents a hierarchical passivity-based compliance controller that exploits robot redundancy, and aims at achieving an impedance behavior with a time-varying stiffness on all priority levels. Unfortunately, this gives rise to certain control actions that lead to the loss of the safety-critical passivity feature. To deal with this problem, we employ the concept of virtual *Energy tanks* that keep track of the passivity-violating energy in the system, ensuring that it remains bounded. This restores the passivity in the system, which guarantees the stable interaction with any passive environment. First, the controller is formulated for the simple case where a hirearchy with two tasks is considered, and subsequently extended for an arbitrary number of prioritized tasks. Furthermore, we augment our controller with an additional safety layer which ensures that the energy injected through the tank into the system remains below a safe limit, defined based on the maximum kinetic energy allowed in the system. Finally, our approach is validated in terms of performance during task execution and safety both in simulations and on real robot hardware. The contents of this chapter are based on the following publications [17,22]

Y. Michel, C. Ott and D. Lee, "Passivity-based variable impedance control for redundant manipulators", in IFAC-PapersOnLine, pp. 9865-9872, 2020.

Y. Michel, C. Ott and D. Lee, "Safety-Aware Hierarchical Passivity-Based Variable Compliance Control for Redundant Manipulators," in IEEE Transactions on Robotics, vol. 38, no. 6, pp. 3899-3916, 2022.

3.1 Motivation

While variable impedance control can endow robots with enhanced adaptation capabilities, stiffness variations generally do not constitute a passive action, which in

consequence, could endanger the stability of the robot. As reviewed earlier (section 2.1.2), several works have aimed to restore the passivity for a variable impedance controlled robot, for example by augmenting the system with an energy tank [96], or deriving in advance conditions on the variable impedance profile to ensure a stable behavior [114]. Surprisingly however, the aspect of robot redundancy is often neglected in VIC. The work in [117] considered redundancy in VIC to enlarge the range of stable impedance parameters, however without explicitly specifying a desired compliance behavior for the null space. Furthermore, the robot was admittance controlled for the purposes of a human-robot collaboration scenario.
In this chapter, we extend the hierarchical compliance control framework, which was initially proposed in [224] for the regulation case and later extended in [225] for an arbitrary number of prioritized tasks, to further include variable stiffness behaviors. In contrast to previous works where stiffness variations are considered only for the main end-effector task [96, 114], we present a controller that takes into account the redundant degrees of freedom, and aims at achieving an impedance behavior with a time-varying stiffness, for an arbitrary number of tasks prioritized in a strict hierarchy. The null-space projections used to implement the hierarchy together with stiffness variations are two non-passive control actions that represent a source of activity in the system [12, 96], jeopardizing the valuable passivity feature. Similar to the works in section 2.1.4, we also leverage the concept of energy tanks to preserve passivity in the system by compensating for the non-passive control actions. While [154], [12] and [96] considered energy tanks for the passivity loss resulting from stiffness variations and null space projections separately, our desired goal of achieving a hierarchical variable compliance controller on all hierarchy levels requires a tank design that takes into account and treats these two problems simultaneously. Furthermore, we analyze how the design of the tank can be related to a concrete safety metric. Our work draws inspiration from the aforementioned literature in section 2.1.4, and also considers energy as a safety indicator in the system. More specifically, we relate the design criteria of our energy tank, used to preserve passivity, to the maximum allowable kinetic energy in the system. To this end, we investigate two possible solutions that can achieve our desiderata. The first solution relies on the choice of the initial tank energy based on the maximum allowed kinetic energy in the system. While simple, we experimentally demonstrate that this solution can restrict the performance, and therefore proceed to present a more efficient solution that limits the power the passivity violating ports can inject into the system. With this additional safety layer, we ensure that the energy these ports can introduce into the system will always remain below a safe limit, in addition to staying bounded, which is already guaranteed by the presence of an upper energy limit to the tank. To summarize, we present the following contributions:

- A *time-varying* compliance controller formulation for the regulation case, initially developed for the case of a two-task hierarchy, and extended to include an arbitrary number of prioritized tasks.

- Augmenting the tank with an additional layer to enforce kinetic energy con-

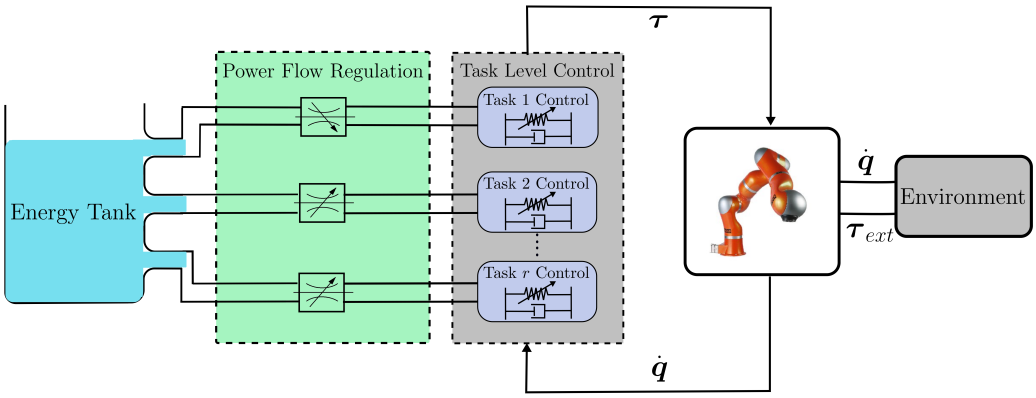


Fig. 3.1: Conceptual Illustration of the overall system architecture, highlighting the different components involved in our approach. ©2022 IEEE [17]

straints, further enhancing the safety of our controller.

- Extensive validations in simulations and on real robot hardware.

A conceptual illustration of our framework is shown in Figure 3.1.

3.2 Control Design for a Two-Task Hierarchy

This section introduces the controller for the case of two tasks in the hierarchy. First, the used dynamic model is explained, and then classical hierarchical impedance controller is discussed, revealing the causes for the passivity loss. Then, we introduce the sought solution highlighting how energy tanks are used to restore passivity in the system. Finally, simulation results validate the proposed theoretical findings.

3.2.1 Dynamic Modeling

In this section, we consider a task hierarchy consisting of two priority levels with a primary task $\boldsymbol{x}_1$ and a null space task $\boldsymbol{x}_2$, which shall be executed as good as possible without disturbing the main task. The tasks are given by $\boldsymbol{x}_i = \boldsymbol{f}_i(\boldsymbol{q})$, where $\boldsymbol{f}_i(\boldsymbol{q})$ is the nonlinear forward kinematic map for $i = 1, 2$. On a differential level, the tasks are defined by:

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{J}_1(\boldsymbol{q})\dot{\boldsymbol{q}} \ , \quad \boldsymbol{J}_1(\boldsymbol{q}) = \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{q}} \tag{3.1}$$

$$\dot{\boldsymbol{x}}_2 = \boldsymbol{J}_2(\boldsymbol{q})\dot{\boldsymbol{q}} \ , \quad \boldsymbol{J}_2(\boldsymbol{q}) = \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{q}} \tag{3.2}$$

where $\boldsymbol{J}_1 \in \mathbb{R}^{m_1 \times n}$ and $\boldsymbol{J}_2 \in \mathbb{R}^{m_2 \times n}$ are the corresponding task Jacobians relating joint-space velocities to task-space velocities. However, since the task space velocities

$\dot{\boldsymbol{x}}_1$, $\dot{\boldsymbol{x}}_2$ feature couplings between the two priority levels, we consider the following coordinate transformation [226]

$$\dot{\boldsymbol{q}} = [\boldsymbol{J}_1(\boldsymbol{q})^{\boldsymbol{W}+} \quad \boldsymbol{Z}(\boldsymbol{q})^T] \begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \boldsymbol{v}_n \end{bmatrix}, \tag{3.3}$$

where $\boldsymbol{J}_1(\boldsymbol{q})^{\boldsymbol{W}+}$ is the weighted pseudo inverse of $\boldsymbol{J}_1(\boldsymbol{q})$ with a weight $\boldsymbol{W}(\boldsymbol{q})$[1] i.e $\boldsymbol{J}_1^{\boldsymbol{W}+} = \boldsymbol{W}^{-1}\boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^T)^{-1}$, $\boldsymbol{Z}(\boldsymbol{q}) \in \mathbb{R}^{(n-m_1)\times n}$ is a full row rank null space base matrix of $\boldsymbol{J}_1(\boldsymbol{q})$ and can be found via singular value decomposition of $\boldsymbol{J}_1(\boldsymbol{q})$. The minimal parameterization of the weighted self motion velocity $\boldsymbol{v}_n \in \mathbb{R}^{m_2\times n}$ is given by

$$\boldsymbol{v}_n = \boldsymbol{N}(\boldsymbol{q})\dot{\boldsymbol{q}}\,, \tag{3.4}$$

with $\boldsymbol{N}(\boldsymbol{q}) = (\boldsymbol{Z}\boldsymbol{W}\boldsymbol{Z}^{\mathsf{T}})^{-1}\boldsymbol{Z}\boldsymbol{W}$ [2]. We can now define the extended Jacobian of the system as

$$\boldsymbol{J}_e(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_1(\boldsymbol{q}) \\ \boldsymbol{N}(\boldsymbol{q}) \end{bmatrix}, \; \boldsymbol{J}_e(\boldsymbol{q})^{-1} = \begin{bmatrix} \boldsymbol{J}_1^{\boldsymbol{W}+}(\boldsymbol{q}) & \boldsymbol{Z}(\boldsymbol{q})^T \end{bmatrix} \tag{3.5}$$

and the extended space velocities

$$\dot{\boldsymbol{x}}_e = \begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \boldsymbol{v}_n \end{bmatrix} = \boldsymbol{J}_e\dot{\boldsymbol{q}}\,, \tag{3.6}$$

which are kinematically decoupled in the sense that $\boldsymbol{J}_1\boldsymbol{Z}^T\boldsymbol{v}_n = \boldsymbol{0}$ and $\boldsymbol{N}\boldsymbol{J}_1^{\boldsymbol{W}+}\dot{\boldsymbol{x}}_1 = \boldsymbol{0}$. Assigning $\boldsymbol{W}(\boldsymbol{q}) = \boldsymbol{M}(\boldsymbol{q})$ and projecting the dynamics (2.10) into extended velocity space yields the dynamically consistent formulation of the hierarchical dynamics [224]

$$\boldsymbol{\Lambda}_e(\boldsymbol{q})\ddot{\boldsymbol{x}}_e + \boldsymbol{\mu}_e(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{x}}_e = \boldsymbol{J}_e(\boldsymbol{q})^{-T}(-\boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau} + \boldsymbol{\tau}_{ext})\,, \tag{3.7}$$

where

$$\boldsymbol{\Lambda}_e = \boldsymbol{J}_e^{-T}\boldsymbol{M}\boldsymbol{J}_e^{-1} = \begin{bmatrix} \boldsymbol{\Lambda}_x(\boldsymbol{q}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Lambda}_n(\boldsymbol{q}) \end{bmatrix}, \tag{3.8}$$

is the extended space inertia which is block diagonal thanks to the particular choice $\boldsymbol{W}(\boldsymbol{q}) = \boldsymbol{M}(\boldsymbol{q})$, which implies that no kinetic energy couplings exist between the priority levels. The extended space Coriolis and centrifugal matrix $\boldsymbol{\mu}_e$ is given by

$$\boldsymbol{\mu}_e = \boldsymbol{\Lambda}_e(\boldsymbol{J}_e\boldsymbol{M}^{-1}\boldsymbol{C} - \dot{\boldsymbol{J}}_e)\boldsymbol{J}_e^{-1} = \begin{bmatrix} \boldsymbol{\mu}_x(\boldsymbol{q},\dot{\boldsymbol{q}}) & \boldsymbol{\mu}_{xn}(\boldsymbol{q},\dot{\boldsymbol{q}}) \\ \boldsymbol{\mu}_{nx}(\boldsymbol{q},\dot{\boldsymbol{q}}) & \boldsymbol{\mu}_n(\boldsymbol{q},\dot{\boldsymbol{q}}) \end{bmatrix}. \tag{3.9}$$

Similar to [224], we can define the control torque $\boldsymbol{\tau}$ as

$$\boldsymbol{\tau} = \boldsymbol{\tau}_g + \boldsymbol{\tau}_d + \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2\,, \tag{3.10}$$

where $\boldsymbol{\tau}_g = \boldsymbol{g}(\boldsymbol{q})$ is a gravity compensation term. The control $\boldsymbol{\tau}_d = \boldsymbol{J}^T\boldsymbol{\mu}_{xn}\boldsymbol{v}_n + \boldsymbol{N}^T\boldsymbol{\mu}_{nx}\dot{\boldsymbol{x}}_1$ is a passive feedback with power $P_d = \boldsymbol{\tau}_d^T\dot{\boldsymbol{q}} = 0$ since $\boldsymbol{\mu}_{nx} = -\boldsymbol{\mu}_{xn}^T$, and

[1] Assigning $\boldsymbol{W}(\boldsymbol{q}) = \boldsymbol{I}$ provides the standard pseudo inverse. In this work, as shown later, the choice $\boldsymbol{W}(\boldsymbol{q}) = \boldsymbol{M}(\boldsymbol{q})$ is made to achieve dynamic consistency

[2] For the sake of clarity, we omit dependency on $\boldsymbol{q}$ where obvious.

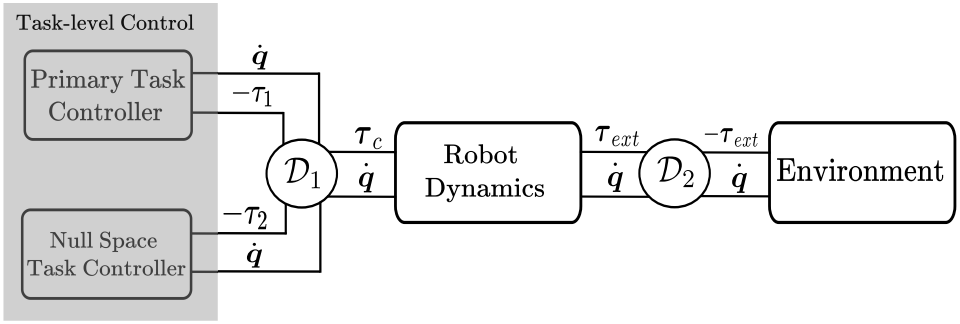


Fig. 3.2: Port-Based modeling of the system as an interconnection of the controllers, the robot dynamics (the plant) and the environment.

has the role of removing the cross coupling Coriolis and centrifugal terms. The torques $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ will be designed to perform the control objective of the primary and null space tasks, respectively. We can now write the extended space dynamics as

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\Lambda}_x & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_n \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{v}}_n \end{bmatrix} + \begin{bmatrix} \boldsymbol{\mu}_x & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mu}_n \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \boldsymbol{v}_n \end{bmatrix} \\ = \boldsymbol{J}_e^{-T} (\underbrace{\boldsymbol{\tau}_1 + \boldsymbol{\tau}_2}_{\boldsymbol{\tau}_c} + \boldsymbol{\tau}_{ext}) \,, \end{aligned} \tag{3.11}$$

with the skew symmetry of $(\dot{\boldsymbol{\Lambda}}_x - 2\boldsymbol{\mu}_x)$ and $(\dot{\boldsymbol{\Lambda}}_n - 2\boldsymbol{\mu}_n)$.

3.2.2 Subsystem Passivity

In order to perform the preliminary passivity analysis of the system, we take inspiration from the *control as interconnection* paradigm [221] in order to simplify the forthcoming analysis. More specifically, a controller can be seen as a dynamical system with its own structure and power ports, and that shall be interconnected to the plant in a power-preserving manner. Therefore, and relying on the fact that the interconnection of passive subsystems yields a passive system, we can study the passivity of the overall system by analyzing the passivity of simpler subsystems independently with respect to their corresponding power ports. As shown in Figure 3.2, the robot dynamics (3.11) is interconnected through the power port $(\boldsymbol{\tau}_{ext}, \dot{\boldsymbol{q}})$ which models the physical interaction with the environment. On the other hand, the port $(\boldsymbol{\tau}_c, \dot{\boldsymbol{q}})$ models the power exchange between the robot and the primary and null space tasks controllers, with the energy injected through this port is used for achieving the control objectives. The topology of the interconnections is defined by the power preserving *Dirac structures* $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ which simply represent a feedback interconnection.

Robot Dynamics

The passivity of the extended task space dynamics (3.11) can be easily shown with the storage function

$$S_r = \frac{1}{2}\dot{\boldsymbol{x}}_e^T \boldsymbol{\Lambda}_e \dot{\boldsymbol{x}}_e \,, \tag{3.12}$$

which represents the sum of kinetic energies of the main and null space tasks. The time derivative of S_r along (3.11) is given by

$$\dot{S}_r = \dot{\boldsymbol{x}}_e^T \boldsymbol{J}_e^{-T}(\boldsymbol{\tau}_c + \boldsymbol{\tau}_{ext}) = \dot{\boldsymbol{q}}^T(\boldsymbol{\tau}_c + \boldsymbol{\tau}_{ext}) \,, \tag{3.13}$$

which following definition 4, shows the passivity of the robot dynamics with respect to the pair $(\boldsymbol{\tau}_c + \boldsymbol{\tau}_{ext}, \dot{\boldsymbol{q}})$.

Primary task controller

The goal of the primary task is to achieve a specific compliance around a virtual equilibrium $\boldsymbol{x}_{1,d}$. Compliance control is a special case of impedance control where inertia shaping and hence the feedback of external forces are avoided. The compliance behavior is defined by a time-varying stiffness matrix $\boldsymbol{K}_1(t)$ and a damping behavior given by a positive definite matrix $\boldsymbol{D}_1$. The classical compliance controller for the primary task can be designed as

$$\boldsymbol{\tau}_1 = \boldsymbol{J}_1^T \boldsymbol{F}_1 \,, \tag{3.14}$$

with the control force $\boldsymbol{F}_1$

$$\boldsymbol{F}_1 = -\boldsymbol{K}_1(t)\tilde{\boldsymbol{x}}_1 - \boldsymbol{D}_1 \dot{\boldsymbol{x}}_1 \,, \tag{3.15}$$

where $\tilde{\boldsymbol{x}}_1 = \boldsymbol{x}_1 - \boldsymbol{x}_{1,d}$. In order to analyze the passivity of this control law, we consider as storage function the associated spring potential

$$S_p = \frac{1}{2}\tilde{\boldsymbol{x}}_1^T \boldsymbol{K}_1(t)\tilde{\boldsymbol{x}}_1 \,. \tag{3.16}$$

Using (3.15), the time derivative of S_p becomes

$$\begin{aligned} \dot{S}_p &= \dot{\boldsymbol{x}}_1^T \boldsymbol{K}_1(t)\tilde{\boldsymbol{x}}_1 + \frac{1}{2}\tilde{\boldsymbol{x}}_1^T \dot{\boldsymbol{K}}_1(t)\tilde{\boldsymbol{x}}_1 \\ &= -\dot{\boldsymbol{x}}_1^T \boldsymbol{F}_1 - \dot{\boldsymbol{x}}_1^T \boldsymbol{D}_1 \dot{\boldsymbol{x}}_1 + \frac{1}{2}\tilde{\boldsymbol{x}}_1^T \dot{\boldsymbol{K}}_1(t)\tilde{\boldsymbol{x}}_1 \end{aligned} \tag{3.17}$$

where passivity with respect to the pair $(-\boldsymbol{F}_1, \dot{\boldsymbol{x}}_1)$ implies passivity with respect to $(-\boldsymbol{\tau}_1, \dot{\boldsymbol{q}})$ since the operator $\boldsymbol{J}$ acts dually in the effort and flow paths. However, passivity of the control law cannot be guaranteed since, although the sign of the dissipated power $\dot{\boldsymbol{x}}_1^T \boldsymbol{D}_1 \dot{\boldsymbol{x}}_1$ is positive semi-definite, the sign of the last term resulting from the stiffness variations is indefinite, and accordingly the primary task controller may be active during periods with increasing stiffness.

Null space controller

Similar to the primary task, the goal of the null space controller is to achieve a specific compliance behavior around a virtual equilibrium $\boldsymbol{x}_{2,d}$ with a stiffness $\boldsymbol{K}_2(t)$ and positive definite damping $\boldsymbol{D}_2$. Nevertheless, this goal shall be achieved only as good as possible without disturbing the primary task. We consider the following control law [224]

$$\boldsymbol{\tau}_2 = \boldsymbol{P}\boldsymbol{J}_2^T\boldsymbol{F}_2 \,, \tag{3.18}$$

with $\boldsymbol{F}_2 = -\boldsymbol{K}_2(t)\tilde{\boldsymbol{x}}_2 - \boldsymbol{D}_2\dot{\boldsymbol{x}}_2$, the task space error $\tilde{\boldsymbol{x}}_2 = \boldsymbol{x}_2 - \boldsymbol{x}_{2,d}$ and $\boldsymbol{P} = \boldsymbol{N}(\boldsymbol{q})^T\boldsymbol{Z}(\boldsymbol{q})$ projects into the null space of the primary task. Unfortunately, as explained in [149, 154], the use of this projection results in a loss of passivity even for a constant stiffness $\boldsymbol{K}_2$, considering that $\boldsymbol{P}$ acts only on force and not dually on velocity. In fact, passivity can be restored if one uses $\dot{\tilde{\boldsymbol{x}}}_2 = \boldsymbol{J}_2\boldsymbol{P}^T\dot{\boldsymbol{q}}$ and its integration for the implementation of the control law (3.18). However since $\dot{\boldsymbol{x}}_2$ is given by

$$\begin{aligned}\dot{\boldsymbol{x}}_2 &= \boldsymbol{J}_2\dot{\boldsymbol{q}} \\ &= \boldsymbol{J}_2(\boldsymbol{J}_1(\boldsymbol{q})^{\boldsymbol{W}+}\dot{\boldsymbol{x}}_1 + \boldsymbol{Z}^T\boldsymbol{v}_n)\,,\end{aligned} \tag{3.19}$$

recalling that $\boldsymbol{v}_n = \boldsymbol{N}(\boldsymbol{q})\dot{\boldsymbol{q}}$, (3.19) becomes

$$\dot{\boldsymbol{x}}_2 = \underbrace{\boldsymbol{J}_2\boldsymbol{J}_1(\boldsymbol{q})^{\boldsymbol{W}+}\dot{\boldsymbol{x}}_1}_{\boldsymbol{\xi}} + \underbrace{\boldsymbol{J}_2\boldsymbol{P}^T\dot{\boldsymbol{q}}}_{\dot{\tilde{\boldsymbol{x}}}_2}\,, \tag{3.20}$$

which means that $\dot{\boldsymbol{x}}_2$ features couplings ($\boldsymbol{\xi}$) from the primary task and therefore, using $\dot{\tilde{\boldsymbol{x}}}_2$ instead of $\dot{\boldsymbol{x}}_2$ would not represent anymore the goal of the compliance controller for the null space task. The passivity of the null space control action (3.18) can be analyzed with the following storage function

$$S_n = \frac{1}{2}\tilde{\boldsymbol{x}}_2^T\boldsymbol{K}_2(t)\tilde{\boldsymbol{x}}_2 \tag{3.21}$$

which also represents the spring potential energy of the null space controller. Its time derivative is given by

$$\dot{S}_n = \dot{\boldsymbol{x}}_2^T\boldsymbol{K}_2(t)\tilde{\boldsymbol{x}}_2 + \frac{1}{2}\tilde{\boldsymbol{x}}_2^T\dot{\boldsymbol{K}}_2(t)\tilde{\boldsymbol{x}}_2 \tag{3.22}$$

and using (3.18), (3.20)

$$\begin{aligned}\dot{S}_n &= (\boldsymbol{\xi}^T + \dot{\tilde{\boldsymbol{x}}}_2^T)(-\boldsymbol{F}_2 - \boldsymbol{D}_2\dot{\boldsymbol{x}}_2) + \frac{1}{2}\tilde{\boldsymbol{x}}_2^T\dot{\boldsymbol{K}}_2(t)\tilde{\boldsymbol{x}}_2 \\ &= -\dot{\boldsymbol{q}}^T\boldsymbol{\tau}_2 - \dot{\boldsymbol{x}}_2^T\boldsymbol{D}_2\dot{\boldsymbol{x}}_2 - \boldsymbol{\xi}^T\boldsymbol{F}_2 + \frac{1}{2}\tilde{\boldsymbol{x}}_2^T\dot{\boldsymbol{K}}_2(t)\tilde{\boldsymbol{x}}_2\end{aligned} \tag{3.23}$$

where the sign of the last two terms cannot be determined implying that the null space controller can be active. Note that, if (3.18) uses as a state $\bar{\boldsymbol{x}}_2$ instead of $\boldsymbol{x}_2$, the term $\boldsymbol{\xi}^T\boldsymbol{F}_2$ vanishes with $\dot{S}_n$ reducing to

$$\dot{S}_n = -\dot{\boldsymbol{q}}^T\boldsymbol{\tau}_2 - \dot{\tilde{\boldsymbol{x}}}_2^T\boldsymbol{D}_2\dot{\tilde{\boldsymbol{x}}}_2 + \frac{1}{2}\tilde{\bar{\boldsymbol{x}}}_2^T\dot{\boldsymbol{K}}_2(t)\tilde{\bar{\boldsymbol{x}}}_2 \tag{3.24}$$

with $S_n = \frac{1}{2}\tilde{\bar{\boldsymbol{x}}}_2^T\boldsymbol{K}_2(t)\tilde{\bar{\boldsymbol{x}}}_2$ as spring potential and error state $\tilde{\bar{\boldsymbol{x}}}_2 = \bar{\boldsymbol{x}}_2 - \boldsymbol{x}_{2,d}$. (3.24) means that the controller would be passive with respect to $(-\boldsymbol{\tau}_2, \dot{\boldsymbol{q}})$ for a constant stiffness ($\dot{\boldsymbol{K}}_2(t) = \boldsymbol{0}$).

3.2.3 Control Design

As shown in the previous section, the primary and null space tasks controllers can become active, and as a result, there exists a passive environment that could destabilize the system when interconnected to the robot [149]. Such a consequence is a major safety issue and is certainly not desirable for robots operating close to humans. The loss of passivity in the system can be stated due to the following reasons:

- Stiffness variations in the main task controller
- Stiffness variations in the null space task controller
- Projection operator in the null space task controller

In order to restore passivity in the system, we employ the concept of *Energy tanks*. We propose a tank design that implements a hybrid impedance-admittance structure, and simultaneously treats the three aforementioned reasons of activity in the system.

Controllers modification

We start by modifying the primary task controller $\boldsymbol{F}_1$ as

$$\boldsymbol{F}_1 = -\boldsymbol{K}_{1,c}\tilde{\boldsymbol{x}}_1 - \boldsymbol{D}_1\dot{\boldsymbol{x}}_1 + \boldsymbol{F}_{1,d}\,, \tag{3.25}$$

and for the null space controller

$$\boldsymbol{F}_2 = -\boldsymbol{K}_{2,c}\tilde{\check{\boldsymbol{x}}}_2 - \boldsymbol{D}_2\dot{\check{\boldsymbol{x}}}_2 + \boldsymbol{F}_{2,d}\,, \tag{3.26}$$

where $\boldsymbol{K}_1 = \boldsymbol{K}_{1,c}+\boldsymbol{K}_{1,d}(t)$ and $\boldsymbol{K}_2 = \boldsymbol{K}_{2,c}+\boldsymbol{K}_{2,d}(t)$. The terms $\boldsymbol{K}_{1,c}$, $\boldsymbol{K}_{2,c}$ represent the constant stiffness part while $\boldsymbol{K}_{1,d}$, $\boldsymbol{K}_{2,d}$ represent the time-varying part. The control actions $\boldsymbol{F}_{1,d}$, $\boldsymbol{F}_{2,d}$ will be used to perform stiffness variations in a passive manner. Furthermore, the new task error is given by $\tilde{\check{\boldsymbol{x}}}_2 = \check{\boldsymbol{x}}_2 - \boldsymbol{x}_{2,d}$ with the new state $\check{\boldsymbol{x}}_2$, which results from integrating

$$\dot{\check{\boldsymbol{x}}}_2 = \boldsymbol{\nu} + \dot{\tilde{\boldsymbol{x}}}_2\,, \tag{3.27}$$

where the auxiliary control input $\boldsymbol{\nu}$ has the role of preserving the original controller state $\dot{\boldsymbol{x}}_2$ as good as possible while ensuring passivity in the system. The forces $\boldsymbol{F}_{1,d}$, $\boldsymbol{F}_{2,d}$ and $\boldsymbol{\nu}$ will be further derived in the following.

Energy Tanks

As highlighted earlier, the introduction of the energy tank will be used for restoring passivity in the system. As explained in sec. 2.2.2, the tank can be viewed as a virtual storage element with passive dynamics interconnected to the controllers in a power preserving manner and will be used to monitor the consequence of non-passive control actions. As long as the tank is not empty, the energy left in the tank will be used to "balance" such actions and therefore ensuring passivity.

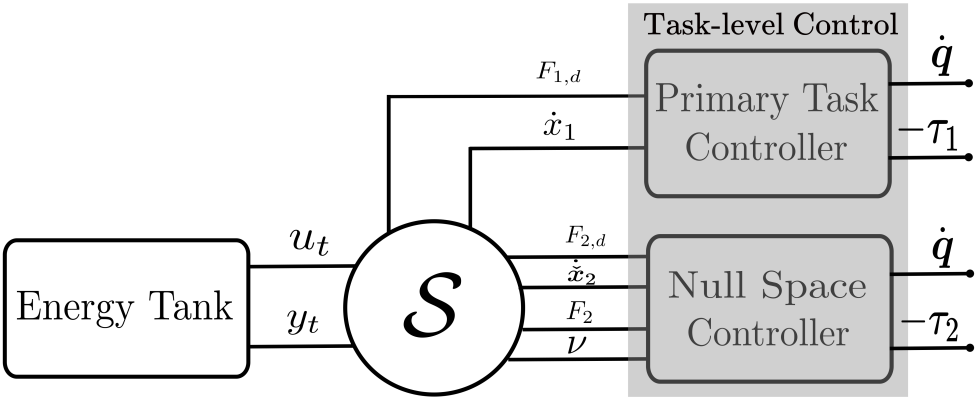


Fig. 3.3: Interconnection of the energy tank with the primary, null space tasks controllers. The gray block Task level Control corresponds to the same block in Fig. 3.2.

We define a tank with a state $x_t \in \mathbb{R}$, an output variable $y_t \in \mathbb{R}$ and an input variable $u_t \in \mathbb{R}$. The tank dynamics is given by

$$\begin{aligned} \dot{x}_t &= \frac{\beta_1 \dot{\boldsymbol{x}}_1^T \boldsymbol{D}_1 \dot{\boldsymbol{x}}_1}{x_t} + \frac{\beta_2 \dot{\tilde{\boldsymbol{x}}}_2^T \boldsymbol{D}_2 \dot{\tilde{\boldsymbol{x}}}_2}{x_t} + u_t \\ y_t &= x_t \end{aligned} \tag{3.28}$$

The energy stored in the tank is given by

$$E_t = \frac{1}{2} x_t^2 \tag{3.29}$$

The tank exchanges energy with the controllers through the power port (u_t, y_t). The terms β_1 and β_2 control the amount of energy stored in the tank through dissipation. They are designed as

$$\beta_i = \begin{cases} \kappa_i, & \text{if } E_t < \bar{E}_t \,, \\ 0, & \text{otherwise} \end{cases} \tag{3.30}$$

for $i = 1, 2$ and with $\kappa_i \in [0, 1]$. The quantity $\bar{E}_t$ is an adequate upper limit for energy stored in the tank. As noted in [227], practically unstable control actions could be masked if the energy stored in the tank is not bounded. Note that, if β_1 and β_2 are set permanently to zero, no refilling of the tank is allowed. This means, that as soon as the initial allocated energy budget is consumed, active control actions generating energy could be no longer tolerated.
The tank is interconnected to the controllers through the Dirac structure

$$\begin{bmatrix} \boldsymbol{F}_{1,d} \\ \boldsymbol{F}_{2,d} \\ -\boldsymbol{\nu} \\ u_t \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{w}_1 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{w}_2 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{c} \\ -\boldsymbol{w}_1^T & -\boldsymbol{w}_2^T & -\boldsymbol{c}^T & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{S}} \begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\tilde{\boldsymbol{x}}}_2 \\ \boldsymbol{F}_2 \\ y_t \end{bmatrix}, \tag{3.31}$$

where the matrix $\boldsymbol{S}$ is skew-symmetric which means that the interconnection is power preserving. As shown in Fig.3.3, the primary task controller exchanges energy with the Dirac structure through the power port $(\boldsymbol{F}_{1,d}, \dot{\boldsymbol{x}}_1)$, with the energy injected through this power port is used to implement stiffness variations in the primary task. On the other hand, the null space controller is interconnected to the Dirac structure through two power ports: one $(\boldsymbol{F}_{2,d}, \dot{\check{\boldsymbol{x}}}_2)$ is used to inject energy necessary for the stiffness variations in the null space task, while the other $(\boldsymbol{F}_2, -\boldsymbol{\nu})$ compensates the null space projection. In terms of causality, the tank acts as a hybrid impedance-admittance. The design parameters $\boldsymbol{w}_1$, $\boldsymbol{w}_2$ and $\boldsymbol{c}$ can be regarded as modulating factors that control the power transmission between the tank and the controllers. For $\boldsymbol{w}_1$, the choice

$$\boldsymbol{w}_1 = -\frac{\sigma(E_t)\boldsymbol{K}_{1,d}(t)\tilde{\boldsymbol{x}}_1}{y_t} \tag{3.32}$$

is made, with the valve $\sigma(E_t)$ as

$$\sigma(E_t) = \begin{cases} 0 < \sigma(E_t) \leq 1 & \text{if } E_t > \underline{E}_t\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{3.33}$$

where $\sigma(E_t)$ can be designed to evolve smoothly between 0 and 1 depending on the amount of energy available in the tank. Similarly, the modulating factor $\boldsymbol{w}_2$

$$\boldsymbol{w}_2 = -\frac{\gamma(E_t)\boldsymbol{K}_{2,d}(t)\tilde{\check{\boldsymbol{x}}}_2}{y_t} \tag{3.34}$$

where the valve $\gamma(E_t)$ is given by

$$\gamma(E_t) = \begin{cases} 0 < \gamma(E_t) \leq 1 & \text{if } E_t > \underline{E}_t\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{3.35}$$

which means that, as long as the tank is not depleted, the controllers can be allowed to perform the time-varying stiffness behavior. Note that, in order to avoid singularities, we set a lower limit $\underline{E}_t$ for the energy threshold in the tank. Regarding $\boldsymbol{c}$, it shall be assigned as

$$\boldsymbol{c} = \frac{\alpha(E_t)(\boldsymbol{\xi} + \boldsymbol{\rho}(\boldsymbol{x}_2 - \check{\boldsymbol{x}}_2))}{y_t} \tag{3.36}$$

where the gain $\boldsymbol{\rho}$ is added to avoid drift as done in [154]. As for $\alpha(E_t)$

$$\alpha(E_t) = \begin{cases} 0 < \alpha(E_t) \leq 1 & \text{if } E_t > \underline{E}_t\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{3.37}$$

It could be easily confirmed that, as long as energy is available in the tank, the original task coordinate $\boldsymbol{x}_2$ will be used, achieving the original goal of the null space compliance controller. As soon as the energy is empty, compensation for this non-passive control action is no longer supported, and the state $\check{\boldsymbol{x}}_2$ deviates from the original task coordinate $\boldsymbol{x}_2$ retaining passivity in the system, at the expense of some control performance.

Passivity Proof

As shown earlier, the interconnection between the tank and the controllers is power-continuous, i.e no power is lost along the interconnection due to the lossless Dirac structure with the skew symmetry of the matrix $\boldsymbol{S}$. Therefore, we can analyze the passivity of the subsystem shown in Fig. 3.3, which consists of the tank and the controllers with the combined storage function

$$S_c = \frac{1}{2}\tilde{\boldsymbol{x}}_1^T \boldsymbol{K}_{1,c}\tilde{\boldsymbol{x}}_1 + \frac{1}{2}\tilde{\tilde{\boldsymbol{x}}}_2^T \boldsymbol{K}_{2,c}\tilde{\tilde{\boldsymbol{x}}}_2 + E_t \tag{3.38}$$

where the new coordinate (3.27) has been used to define the spring potential of the null space controller. The time derivative of S_c is given by

$$\dot{S}_c = \dot{\boldsymbol{x}}_1^T \boldsymbol{K}_{1,c}\tilde{\boldsymbol{x}}_1 + \dot{\tilde{\boldsymbol{x}}}_2^T \boldsymbol{K}_{2,c}\tilde{\tilde{\boldsymbol{x}}}_2 + \dot{x}_t x_t \,. \tag{3.39}$$

Using (3.25), (3.26), (3.28) and recalling that $\boldsymbol{F}_{1,d} = \boldsymbol{w}_1 y_t$, $\boldsymbol{F}_{2,d} = \boldsymbol{w}_2 y_t$ from (3.31), we now have

$$\begin{aligned} \dot{S}_c = &-\dot{\boldsymbol{x}}_1^T \boldsymbol{F}_1 - \dot{\tilde{\boldsymbol{x}}}_2^T \boldsymbol{F}_2 - \dot{\boldsymbol{x}}_1^T \boldsymbol{D}_1 \dot{\boldsymbol{x}}_1 - \dot{\tilde{\boldsymbol{x}}}_2^T \boldsymbol{D}_2 \dot{\tilde{\boldsymbol{x}}}_2 \\ &+\dot{\boldsymbol{x}}_1^T \boldsymbol{w}_1 y_t + \dot{\tilde{\boldsymbol{x}}}_2^T \boldsymbol{w}_2 y_t + \beta_1 \dot{\boldsymbol{x}}_1^T \boldsymbol{D}_1 \dot{\boldsymbol{x}}_1 + \beta_2 \dot{\tilde{\boldsymbol{x}}}_2^T \boldsymbol{D}_2 \dot{\tilde{\boldsymbol{x}}}_2 \\ &-\boldsymbol{w}_1^T \dot{\boldsymbol{x}}_1 x_t - \boldsymbol{w}_2^T \dot{\tilde{\boldsymbol{x}}}_2 x_t - \boldsymbol{c}^T \boldsymbol{F}_2 x_t \,. \end{aligned} \tag{3.40}$$

Finally, since $y_t = x_t$ and with $\dot{\tilde{\boldsymbol{x}}}_2 = \boldsymbol{J}_2 \boldsymbol{P}^T \dot{\boldsymbol{q}}$ along with (3.1), (3.14), (3.18) and (3.27), the expression (3.40) simplifies to

$$\dot{S}_c = -\dot{\boldsymbol{q}}^T(\boldsymbol{\tau}_1 + \boldsymbol{\tau}_2) - (1-\beta_1)\dot{\boldsymbol{x}}_1^T \boldsymbol{D}_1 \dot{\boldsymbol{x}}_1 - (1-\beta_2)\dot{\tilde{\boldsymbol{x}}}_2^T \boldsymbol{D}_2 \dot{\tilde{\boldsymbol{x}}}_2 \,. \tag{3.41}$$

Since the last two terms are negative semi-definite for β_1, β_2 defined according to (3.30), passivity of the controllers and the energy tank interconnection is guaranteed with respect to the pair $(-(\boldsymbol{\tau}_1 + \boldsymbol{\tau}_2), \dot{\boldsymbol{q}})$.
Intuitively, this can be interpreted as the passification of the parallel interconnection of the primary and the null space tasks controllers. A more conservative solution, however, is to have two local energy tanks interconnected to each controller independently wherein passivity can be shown with respect to $(-\boldsymbol{\tau}_1, \dot{\boldsymbol{q}})$ and $(-\boldsymbol{\tau}_2, \dot{\boldsymbol{q}})$ for the primary and the null space controller, respectively. In this case, energy exchange across levels will no longer occur, and the energy dissipated by one controller will remain on this level.
Now the following statement can be made about the passivity of the overall system:

Proposition 1 *Consider the system given by* (3.50) *with the closed-loop dynamics* (3.11) *with* $\boldsymbol{\tau}_1$, $\boldsymbol{\tau}_2$ *given by* (3.14), (3.18) *and the corresponding control forces* $\boldsymbol{F}_1$, $\boldsymbol{F}_2$ *defined according to* (3.25), (3.26) *respectively. The controlled robot system defines a passive map with respect to the input* $\boldsymbol{\tau}_{ext}$ *and the output* $\dot{\boldsymbol{q}}$.

The passivity claim follows immediately from the fact that the overall system now is an interconnection of passive subsystems with respect to their corresponding

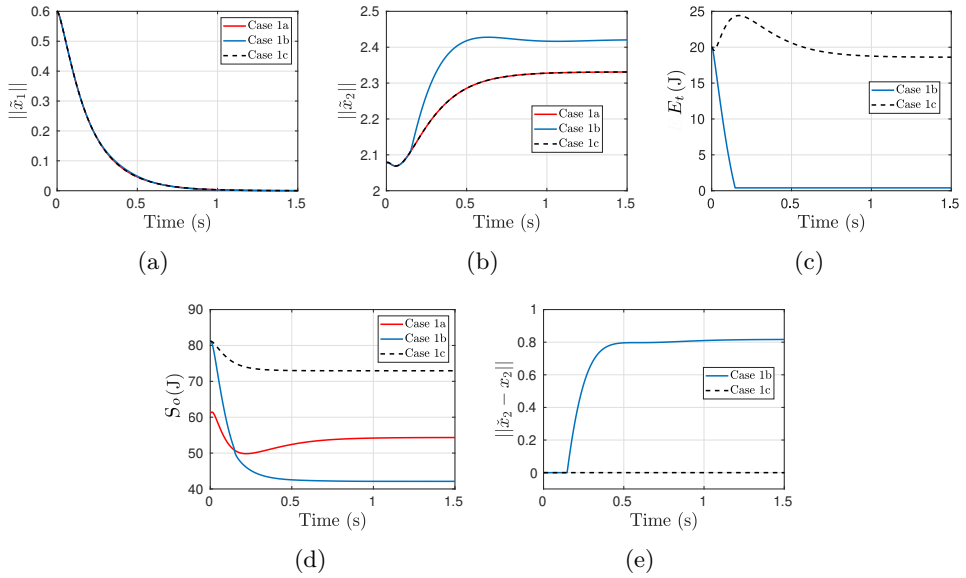


Fig. 3.4: Experimental results for the constant stiffness case, where the errors on levels one and two, the total energy in the system, the tank energy and the deviation between the original and the new coordinates are shown. Legend: Case 1a: Control law with energy tank deactivated. Case 1b : Tank switched on without refilling. Case 1c: Tank switched on with refilling.

input-output pairs, and interconnected through power-conserving Dirac structures. We can consider the following total storage

$$S_o = S_c + S_r \tag{3.42}$$

which is the sum of kinetic, potential and tank energies in the system. Its rate of change is given by

$$\begin{aligned} \dot{S}_o &= \dot{S}_c + \dot{S}_r \\ &= -\dot{\boldsymbol{q}}^T \boldsymbol{\tau}_c - (1-\beta_1)\dot{\boldsymbol{x}}_1^T \boldsymbol{D}_1 \dot{\boldsymbol{x}}_1 - (1-\beta_2)\dot{\tilde{\boldsymbol{x}}}_2^T \boldsymbol{D}_2 \dot{\tilde{\boldsymbol{x}}}_2 \\ &\quad + \dot{\boldsymbol{q}}^T (\boldsymbol{\tau}_c + \boldsymbol{\tau}_{ext}) \\ &= \dot{\boldsymbol{q}}^T \boldsymbol{\tau}_{ext} - (1-\beta_1)\dot{\boldsymbol{x}}_1^T \boldsymbol{D}_1 \dot{\boldsymbol{x}}_1 - (1-\beta_2)\dot{\tilde{\boldsymbol{x}}}_2^T \boldsymbol{D}_2 \dot{\tilde{\boldsymbol{x}}}_2 \end{aligned} \tag{3.43}$$

which proves passivity with respect to the port $(\boldsymbol{\tau}_{ext}, \dot{\boldsymbol{q}})$, through which the controlled robot interacts with the environment, which ensures a stable interaction with any passive environment [149].

3.2.4 Simulations

In order to validate the presented approach, simulations were performed on a 4R-DOF planar manipulator. Each link has 0.5 m length and a 0.5 kg point mass located at the center. The initial configuration is $\boldsymbol{q}_0 = [135°, -90°, -45°, -45°]^T$. The primary

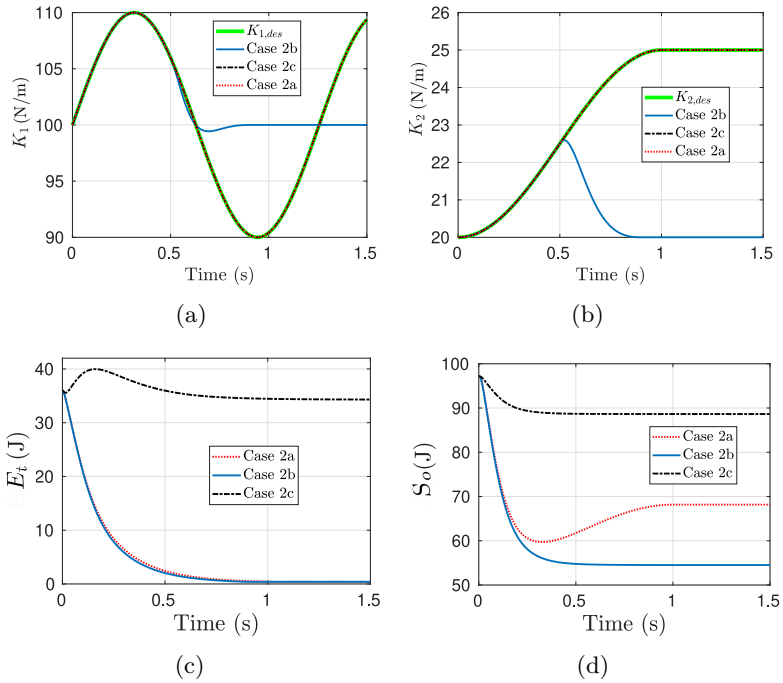


Fig. 3.5: Results with the time-varying stiffness profile. In the upper row, the desired and actual stiffnesses for the primary and null space tasks are shown. In the bottom row, the level of energy in the tank and the total energy in the system are depicted. Legend: Case 2a: Tank switched on without stiffness compensation. Case 2b: Tank switched on with stiffness compensation but without refilling. Case 2c: Tank switched on with stiffness compensation and with refilling.

task is an X/Y Cartesian impedance control at the end effector ($\boldsymbol{x}_1 \in \mathbb{R}^2$), while the null space task is a joint level impedance control ($\boldsymbol{x}_2 \in \mathbb{R}^4$). The desired equilibrium configuration of the joint impedance was deliberately chosen not to be feasible given the constraints imposed by the primary task. Therefore, the null space task is in conflict with the primary task and cannot be fully executed, but only as good as possible without disturbing the primary task.

In the first simulation experiment, the compliance behavior for the two tasks was specified with a constant stiffness. Three scenarios were simulated

- Case 1a : Control law without energy tank
- Case 1b : Control law with energy tank, without refilling, i.e. $\beta_{1,2} = 0$
- Case 1c : Control law with energy tank, with refilling and $\beta_{1,2}$ were set to 0.8

In both the second and third scenarios, the tank is initialized with an energy level of 20 J, however in the second scenario, the dissipated energy by the controllers is completely lost and is not re-used to fill the tank.
The results of this simulation experiment are shown in Fig. 3.4. Concerning the

primary task, it could be observed that the convergence of the error norm $||\tilde{\boldsymbol{x}}_1||$ is identical for all approaches (Fig. 3.4 (a)), since the primary task controller remains unchanged for all scenarios. On the other hand, the task space error norm of the null space task $||\tilde{\boldsymbol{x}}_2||$ never reaches zero due to the conflict between the primary and the null space task, and therefore only convergence to a constrained local minimum is possible. For Case 1a, the steady state error is lower than the one in Case 1b. This occurs due to the deviation between the new passive coordinate $\check{\boldsymbol{x}}_2$ and the original one $\boldsymbol{x}_2$ shown in Fig. 3.4 (e), as soon as the tank gets depleted at $t \approx 0.2$ s, enforcing that the system remains passive at the expense of some control performance. As for Case 1a, the system is clearly not passive as the total energy of the system (Fig. 3.4 (d)) increases temporarily between $t \approx 0.2$ s and $t \approx 0.5$ s. When refilling of the tank is allowed (Case 1c), performance is restored and becomes similar to the classical case, as the tank never gets empty.

In the second simulation experiment, we validate the performance for a time-varying stiffness profile. For the primary task, the following stiffness profile was specified

$$K_{1,des} = 100 + 10sin(5t) \tag{3.44}$$

while for the null space task, the desired stiffness profile is $K_{2,des} = 20 + K_{2,d}(t)$ where $K_{2,d}(t)$ is obtained using the following smoothly rising function

$$\mathcal{K}(t,k_f,t_{min},t_{max}) = \begin{cases} 0, & \text{if } t < t_{min}\,, \\ k_f, & \text{if } t > t_{max}, \\ k_f - (\frac{k_f}{2} + \frac{k_f}{2}cos(\frac{\pi(t-t_{min})}{(t_{max}-t_{min})})), & \text{otherwise.} \end{cases} \tag{3.45}$$

parameterized by the time point at the beginning of the transition t_{min}, the time point t_{max} indicating the end of the stiffness increase and the scale k_f for the maximum desired stiffness value, and where the choice $k_f = 5$ was made in this experiment. Three simulation scenarios were tested

- Case 2a : Tank on, without stiffness compensation, i.e σ, γ in (3.32),(3.34) are set always to 1
- Case 2b : Tank on with stiffness compensation, without re-filling with σ, γ set according to (3.33), (3.35) and $\beta_{1,2} = 0$
- Case 2c : Tank on with stiffness compensation, with re-filling with σ, γ set according to (3.33), (3.35) and $\beta_{1,2} = 0.8$

In all scenarios, the tank was initialized with an energy level of 35 J. The results are shown in Fig. 3.5. For Case 2a (shown in red), the desired stiffness profile is respected on both levels, however the system loses the safety-critical passivity feature which can be verified in the evolution of the total energy in the system, as energy increases (Fig. 3.5 (d)). When the tank compensates for stiffness variations, but without re-filling (Case 2b, blue), the time-varying stiffness component can be no longer followed as

soon as the initial allocated energy budget is consumed, sacrificing performance for the sake of preserving passivity. Nevertheless, routing the dissipated energy back to the tank (Case 2c, black) enhances the performance, while still, ensuring the system remains passive and that energy is monotonically decreasing.
While the differences in performance as compared to the classical controller are not major, the passivity-based controller has the advantage of being passive which means that the controlled robot will always yield stable interactions. Furthermore, the developed controller is energy-aware, in the sense that it is possible to assign context and application dependent energy budgets for the execution of certain non-passive control actions. For example, in situations where the robot interacts with humans, a low energy budget can be chosen such that an unexpected contact between a robot and a human would not lead to injuries. Previous works analyzed the maximum amount of energy a human body can sustain [156, 228], and an impedance controller that respects such energy and power limitations was developed in [158]. On the other hand, less conservative energy budgets can be assigned in non-domestic environments where performance would be of higher priority, for instance the accurate reproduction of a time-varying stiffness profile during a contact with a surface.

3.3 Extension to an Arbitrary Number of Tasks

In this section, we extend the foregoing analysis to the case where an arbitrary number of prioritized tasks is considered. Similar to the previous section, we start by formulating the system dynamics and the classical hierarchical controller [225], revealing the potential passivity loss, which is solved by the introduction of the energy tank.

3.3.1 Dynamic Modeling

From hereon, we consider a task hierarchy consisting of an arbitrary number of prioritized tasks r. On a kinematic level, a task $\boldsymbol{x}_i$ with a dimension m_i is defined by $\boldsymbol{x}_i = \boldsymbol{f}_i(\boldsymbol{q})$ where $\boldsymbol{f}_i(.)$ is the forward kinematics map. The tasks are assumed to be partially in conflict, but a lower priority task $\boldsymbol{x}_j$ cannot disturb a higher priority task $\boldsymbol{x}_i$ for $i > j$. On a differential level, the tasks are defined by

$$\dot{\boldsymbol{x}}_i = \boldsymbol{J}_i(\boldsymbol{q})\dot{\boldsymbol{q}} \ , \ \ \boldsymbol{J}_i(\boldsymbol{q}) = \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{q}} \tag{3.46}$$

where $\boldsymbol{J}_i \in \mathbb{R}^{m_i \times n}$ is the corresponding task Jacobian. For formulating the hierarchical controllers, we replace the task velocities $\dot{\boldsymbol{x}}_1 \dots \dot{\boldsymbol{x}}_r$ that feature couplings between the priority levels [224], with the locally decoupled velocities $\boldsymbol{v}_i \in \mathbb{R}^{m_i} \, \forall \, 1 \leq i < r$ and $\boldsymbol{v}_i \in \mathbb{R}^{n - \sum_{i=1}^{r-1} m_i}$ for $i = r$ through the coordinate transformation [225]

$$\dot{\boldsymbol{x}}_e = \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_r \end{bmatrix} = \boldsymbol{J}_e(\boldsymbol{q})\dot{\boldsymbol{q}} \, , \tag{3.47}$$

where $\dot{\boldsymbol{x}}_e$ is the new extended-space velocity vector formed by stacking the locally decoupled velocities. The mapping $\boldsymbol{J}_e$ denotes the extended Jacobian defined as

$$\boldsymbol{J}_e(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{N}_1(\boldsymbol{q}) \\ \vdots \\ \boldsymbol{N}_r(\boldsymbol{q}) \end{bmatrix} \tag{3.48}$$

with $\boldsymbol{N}_1 = \boldsymbol{J}_1$, $\boldsymbol{N}_i = (\boldsymbol{Z}_i \boldsymbol{M} \boldsymbol{Z}_i^T)^{-1} \boldsymbol{Z}_i \boldsymbol{M}$ $\forall i > 1$ and where $\boldsymbol{Z}_i$ is a full row-rank null space base matrix. It is defined as

$$\boldsymbol{Z}_i(\boldsymbol{q}) = \begin{cases} (\boldsymbol{J}_1(\boldsymbol{q})^{M+})^T & \text{if } i = 1 \\ \boldsymbol{J}_i \boldsymbol{Y}_{i-1}^T (\boldsymbol{Y}_{i-1} \boldsymbol{M} \boldsymbol{Y}_{i-1}^T)^{-1} \boldsymbol{Y}_{i-1} & \text{if } 1 < i < r\,, \\ \boldsymbol{Y}_{r-1} & \text{if } i = r\,. \end{cases} \tag{3.49}$$

where $\boldsymbol{J}_1^{M+} = \boldsymbol{M}^{-1} \boldsymbol{J}_1^T (\boldsymbol{J}_1 \boldsymbol{M}^{-1} \boldsymbol{J}_1^T)^{-1}$ is the inertia weighted pseudo inverse of $\boldsymbol{J}_1$, and where $\boldsymbol{Y}_{i-1}(\boldsymbol{q})$ spans the null space of $[\boldsymbol{J}_1 \dots \boldsymbol{J}_{i-1}]^T$. Using $\dot{\boldsymbol{x}}_e$ as coordinate, we can now project our rigid body dynamics into extended space via $\boldsymbol{J}_e$ obtaining a formulation for the extended space dynamics

$$\boldsymbol{\Lambda}_e(\boldsymbol{q}) \ddot{\boldsymbol{x}}_e + \boldsymbol{\mu}_e(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{x}}_e = \boldsymbol{J}_e(\boldsymbol{q})^{-T} (-\boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau} + \boldsymbol{\tau}_{ext})\,, \tag{3.50}$$

where $\boldsymbol{\Lambda}_e = \boldsymbol{J}_e^{-T} \boldsymbol{M} \boldsymbol{J}_e^{-1}$ is the extended space inertia, and $\boldsymbol{\mu}_e = \boldsymbol{\Lambda}_e (\boldsymbol{J}_e \boldsymbol{M}^{-1} \boldsymbol{C} - \dot{\boldsymbol{J}}_e) \boldsymbol{J}_e^{-1}$ is the extended space Coriolis and centrifugal matrix. Note that now $\boldsymbol{\Lambda}_e$ is block-diagonal thanks to the choice of $\boldsymbol{M}(\boldsymbol{q})$ as weight in the formulation of $\boldsymbol{Z}_i(\boldsymbol{q})$ in (3.49), such that the identity $\boldsymbol{Z}_i \boldsymbol{M} \boldsymbol{Z}_j^T = \boldsymbol{0}$ holds when $i \neq j$, thereby zeroing out the off-diagonal terms [225, 229]. As for $\boldsymbol{\mu}_e(\boldsymbol{q}, \dot{\boldsymbol{q}})$, it still features some off-diagonal couplings, which will be removed via the control action, as originally proposed in [224].

3.3.2 Passivity Analysis

The control action aims at achieving a time-varying compliance behavior on all priority levels, while adhering to a strict hierarchy, where a lower priority task cannot disturb a higher one.
The classical hierarchical compliance control action can be written as [225]

$$\boldsymbol{\tau} = \boldsymbol{\tau}_g + \boldsymbol{\tau}_d + \boldsymbol{\tau}_o \tag{3.51}$$

where $\boldsymbol{\tau}_g$ is as before the gravity cancellation term, and $\boldsymbol{\tau}_d$ is the power-conserving feedback action that compensates for the off-diagonal Coriolis and centrifugal couplings on all priority levels [225]. As for the last control action, it is defined as $\boldsymbol{\tau}_o = \sum_{i=1}^{r} \boldsymbol{\tau}_i$, where the torque $\boldsymbol{\tau}_i$ represents the control action for the execution of the task objective on level i and can be expressed as

$$\boldsymbol{\tau}_i = \boldsymbol{P}_i \boldsymbol{J}_i^T \boldsymbol{F}_i, \tag{3.52}$$

$$\boldsymbol{F}_i = -\boldsymbol{K}_i(t) \tilde{\boldsymbol{x}}_i - \boldsymbol{D}_i(t) \dot{\boldsymbol{x}}_i \tag{3.53}$$

where $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{x}_{i,d}$ is the task space error to a desired static equilibrium $\boldsymbol{x}_{i,d}$, $\boldsymbol{K}_i(t)$ is a stiffness term that consists of a constant positive semi-definite $\boldsymbol{K}_{i,c}$ and a time-varying part $\boldsymbol{K}_{i,d}(t)$ such that $\boldsymbol{K}_i = \boldsymbol{K}_{i,d}(t) + \boldsymbol{K}_{i,c}$. In case such a split is not possible, we can assume that $\boldsymbol{K}_i(t) = \boldsymbol{K}_{i,d}(t)$ and that $\boldsymbol{K}_{i,c} = \boldsymbol{0}$. The matrix $\boldsymbol{D}_i(t)$ is a time-varying damping. Note that, the variation of damping does not have any effect on stability, as long as the matrix remains positive definite. The role of $\boldsymbol{J}_i^T$ is to project the control force into generalized torques using the task Jacobian. As for $\boldsymbol{P}_i$, it represents a projection operator that projects $\boldsymbol{J}_i^T \boldsymbol{F}_i$ into the null space of all higher priority levels. It is defined as

$$\boldsymbol{P}_i(\boldsymbol{q}) = \begin{cases} \boldsymbol{I}_{m_1} & \text{if } i = 1 \\ \boldsymbol{N}_i^T \boldsymbol{Z}_i & \text{otherwise } . \end{cases} \tag{3.54}$$

The passivity of the control action (3.52)-(3.53) can be analyzed with the following storage

$$S = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M} \dot{\boldsymbol{q}} + \sum_{i=1}^{r} \frac{1}{2} \tilde{\boldsymbol{x}}_i^T \boldsymbol{K}_i \tilde{\boldsymbol{x}}_i \tag{3.55}$$

which represents the sum of the total kinetic energy and potential energies from all priority levels. The rate of S along the dynamics (3.50) with $\boldsymbol{\tau}$ defined as in (3.51) is given by

$$\dot{S} = \dot{\boldsymbol{q}}^T \boldsymbol{\tau}_{ext} - \sum_{i=1}^{r} \dot{\boldsymbol{x}}_i^T \boldsymbol{D}_i \dot{\boldsymbol{x}}_i - \sum_{i=2}^{r} \dot{\boldsymbol{\varphi}}_i^T \boldsymbol{F}_i + \sum_{i=1}^{r} \frac{1}{2} \tilde{\boldsymbol{x}}_i^T \dot{\boldsymbol{K}}_{i,d}(t) \tilde{\boldsymbol{x}}_i \tag{3.56}$$

where $\dot{\boldsymbol{\varphi}}_i = \boldsymbol{J}_i \sum_{j=i-1}^{r} \boldsymbol{Z}_j^T \dot{\boldsymbol{v}}_j$. Clearly, the sign of the last two terms is undefined which implies that the system can be active with respect to the input-output pair $(\boldsymbol{\tau}_{ext}, \dot{\boldsymbol{q}})$. The last term results from the stiffness variations, while the third term is due to the use of the projection operator, since $\boldsymbol{P}_i$ does not act dually in the effort and flow paths [12]. One can also restore passivity in the system by using $\dot{\tilde{\boldsymbol{x}}}_i = \boldsymbol{J}_i \boldsymbol{P}_i^T \dot{\boldsymbol{q}}$ as a state for the implementation of the control force $\boldsymbol{F}_i$ in (3.53), but since $\dot{\boldsymbol{x}}_i$ is given by

$$\dot{\boldsymbol{x}}_i = \boldsymbol{J}_i \dot{\boldsymbol{q}} = \underbrace{\boldsymbol{J}_i \sum_{j=i-1}^{r} \boldsymbol{Z}_j^T \dot{\boldsymbol{v}}_j}_{\dot{\boldsymbol{\varphi}}_i} + \underbrace{\boldsymbol{J}_i \boldsymbol{P}_i^T \dot{\boldsymbol{q}}}_{\dot{\tilde{\boldsymbol{x}}}_i}, \tag{3.57}$$

and therefore the use of $\dot{\tilde{\boldsymbol{x}}}_i$ instead of $\dot{\boldsymbol{x}}_i$ does not fully describe the physical state of the task-level controller.

3.3.3 System Passification

In order to restore passivity in the system, energy tanks are also used. While in the previous section, we showed a tank structure with a hybrid impedance-admittance causality, here, a tank structure based on a unified impedance causality is implemented. While practically the performance is similar, the benefit of this representation will become clear in section 3.4.

Controllers Modification

Analogous to the two tasks hierarchy case, we start by modifying the controller for each hierarchy level by splitting the control action into an intrinsically passive part, and non-passive components to be compensated by the tank. The new expression for $\boldsymbol{F}_i$ takes the following form

$$\boldsymbol{F}_i = \boldsymbol{F}_{i,c} + \boldsymbol{F}_{i,d} + \boldsymbol{F}_{i,p} \tag{3.58}$$

where $\boldsymbol{F}_{i,d}$ and $\boldsymbol{F}_{i,p}$ are the potentially passivity violating control actions responsible for the implementation of the time-varying stiffness, and the preservation of the null space projection on level i, respectively, and will be further derived in the following. As for $\boldsymbol{F}_{i,c}$, it represents the intrinsically passive part, and is given by

$$\boldsymbol{F}_{i,c} = -\boldsymbol{K}_{i,c}\tilde{\tilde{\boldsymbol{x}}}_i - \boldsymbol{D}_i\dot{\tilde{\boldsymbol{x}}}_i, \tag{3.59}$$

with the new error state $\tilde{\tilde{\boldsymbol{x}}}_i = \bar{\boldsymbol{x}}_i - \boldsymbol{x}_{i,d}$, and where $\bar{\boldsymbol{x}}_i$ is the result of numerically integrating $\dot{\tilde{\boldsymbol{x}}}_i$ in (3.57). The passivity of this control action can be easily shown by considering the storage

$$S_{i,c} = \frac{1}{2}\tilde{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{K}_{i,c}\tilde{\tilde{\boldsymbol{x}}}_i \tag{3.60}$$

with the rate $\dot{S}_{i,c} = -\dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{F}_{i,c} - \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{D}_i \dot{\tilde{\boldsymbol{x}}}_i$, which proves the passivity of the control action (3.59) with respect to the power port $(-\boldsymbol{F}_{i,c}, \dot{\tilde{\boldsymbol{x}}}_i)$.
It is important to note that the separation of the stiffness control action in (3.58) into $\boldsymbol{F}_{i,c}$ and $\boldsymbol{F}_{i,d}$ is based on the assumption that we can write $\boldsymbol{K}_i = \boldsymbol{K}_{i,d}(t) + \boldsymbol{K}_{i,c}$. We recommend to always seek for such a separation, which has the benefit that the controller becomes less conservative. This is due to the fact that the tank would then compensate only for the passivity-violating control action i.e. the time-varying stiffness behavior.

Energy Tanks

The tank is defined with a state $x_t \in \mathbb{R}$, a storage $E_t = \frac{1}{2}x_t^2$, along with r input and output variables $u_{t,i} \in \mathbb{R}$ and $y_{t,i} \in \mathbb{R}$, respectively, that describe the ports of interaction between the tank and the controllers. The tank dynamics are defined as

$$\dot{x}_t = \frac{\sum_{i=1}^{r} \beta_i \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{D}_i \dot{\tilde{\boldsymbol{x}}}_i}{x_t} - \frac{\sum_{i=1}^{r} u_{t,i} y_{t,i}}{x_t} \tag{3.61}$$

which leads to the following expression of rate of change of tank energy:

$$\dot{E}_t = \sum_{i=1}^{r} \beta_i \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{D}_i \dot{\tilde{\boldsymbol{x}}}_i - \sum_{i=1}^{r} u_{t,i} y_{t,i} \tag{3.62}$$

where β_i controls the amount of energy filled in the tank through dissipation such that

$$\beta_i = \begin{cases} 0 \leq \beta_i \leq 1 & \text{if } E_t < \overline{E}_t \\ 0 & \text{otherwise}\,. \end{cases} \tag{3.63}$$

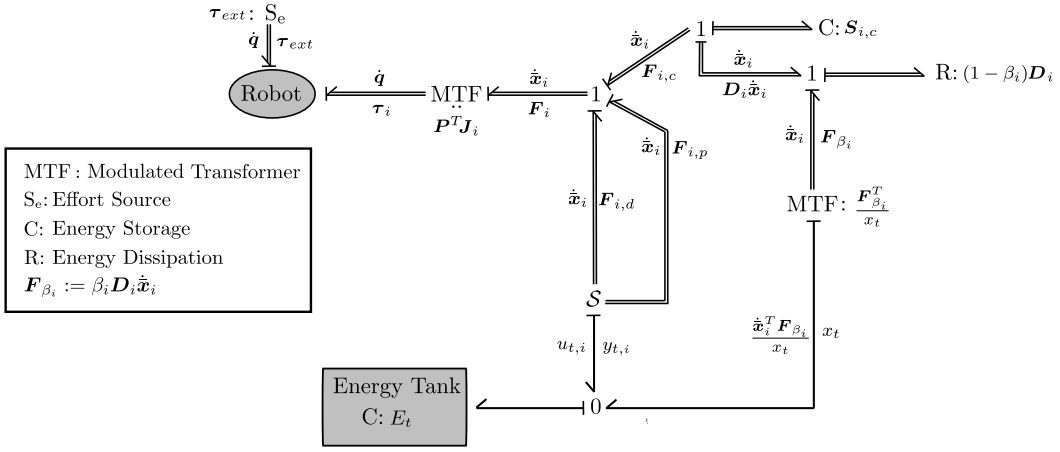


Fig. 3.6: Bond graph model [230] depicting depicting the interconnection of the robot, the controller on level i and the energy tank used to compensate for the non-passive actions and filled through the dissipated energy. ©2022 IEEE [17].

The design choice β_i reflects in some sense the degree of conservatism in the system, as small (large) values mean that the tank gets drained faster (slower), which consequently affects the budget available to compensate for the non-passive actions. As for $\overline{E}_t$, it is the maximum energy allowed in the tank. The tank is interconnected through the port $(u_{t,i}, y_{t,i})$ with the controller of the i-th task with the following Dirac structure

$$\begin{bmatrix} \boldsymbol{F}_{i,d} \\ \boldsymbol{F}_{i,p} \\ u_{t,i} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \frac{\sigma_i \boldsymbol{F}'_{i,d}}{x_t} \\ \mathbf{0} & \mathbf{0} & \frac{\sigma_i \boldsymbol{F}'_{i,p}}{x_t} \\ -\frac{(\sigma_i \boldsymbol{F}'_{i,d})^T}{x_t} & -\frac{(\sigma_i \boldsymbol{F}'_{i,p})^T}{x_t} & \mathbf{0} \end{bmatrix}}_{\boldsymbol{S}} \begin{bmatrix} \dot{\tilde{\boldsymbol{x}}}_i \\ \dot{\tilde{\boldsymbol{x}}}_i \\ y_{t,i} \end{bmatrix}, \tag{3.64}$$

with $y_{t,i} = x_t$, and where the mapping $\boldsymbol{S}$ is skew-symmetric implying that the interconnection is power-conserving.
The force $\boldsymbol{F}'_{i,d}$ is defined as

$$\boldsymbol{F}'_{i,d} = -\boldsymbol{K}_{i,d}(t)(\boldsymbol{x}_i - \boldsymbol{x}_{i,d}), \tag{3.65}$$

while for $\boldsymbol{F}'_{i,p}$

$$\boldsymbol{F}'_{i,p} = \begin{cases} \mathbf{0} & \text{if } i = 1 \\ -\boldsymbol{K}_{i,c}\boldsymbol{\varphi} - \boldsymbol{D}_i\dot{\boldsymbol{\varphi}}_i & \text{if } 1 < i < r\,. \end{cases} \tag{3.66}$$

With the definition of the Dirac structure in (3.64), each task-level controller is interconnected to the tank through two power ports: one port $(\dot{\tilde{\boldsymbol{x}}}_i, \boldsymbol{F}_{i,d})$ where the energy injected through this port is used to implement the time-varying stiffness control action, while the other port $(\dot{\tilde{\boldsymbol{x}}}_i, \boldsymbol{F}_{i,p})$ extracts energy from the tank to preserve the null space projection. Regarding the valve σ_i, it controls the power

transmission between the tank and the controllers, detaching the two from each other as soon as the tank becomes empty. It is assigned as

$$\sigma_i = \begin{cases} 1 & \text{if } E_t > \underline{E_t} \text{ or } \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{F}'_{i,p} + \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{F}'_{i,d} < 0\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{3.67}$$

where $\underline{E_t}$ is a threshold for the minimum energy allowed in the tank to avoid singularities. Through the valve definition in (3.67), one can easily confirm that these potentially non-passive control behaviors can be only implemented as long as 1) some energy remains in the tank or 2) the control behaviors are passive with a negative transmitted power. Our controller architecture for the i-th task is depicted in the form of a bond graph [230] in Figure 3.6.
As stated earlier, the difference compared to the previous section lies in terms of the tank causality, which is reflected in the way the Dirac structure is chosen. For the Dirac structure formulated in (3.64), one can easily see that the tank admits an impedance causality, since the right hand side of the mapping $\boldsymbol{S}$ consists only of a flow variable ($\dot{\tilde{\boldsymbol{x}}}_i$), and outputs the effort variables $\boldsymbol{F}_{i,p}$ and $\boldsymbol{F}_{i,d}$. On the other hand, in (3.31) the right hand side consists of a mixture of flow and effort variables. This has the physical interpretation that in (3.31), the tank aims at preserving the null space projection while guaranteeing passivity by preserving the physical state for the lower priority tasks i.e the flow $\dot{\boldsymbol{\varphi}}_i$, while in (3.64), the tanks aims at the preservation of the impedance control law stemming from the null space projection i.e the effort $\boldsymbol{F}_{i,p}$.

Passivity Analysis

We follow again the *control by interconnection* paradigm [221] to analyze the overall passivity in the system, by diving the system into an interconnection of smaller subsystems. Since the gravity-compensated robot dynamics described by (3.50) with $\boldsymbol{\tau}$ defined according to (3.51) represents a passive map with respect to the port $(\boldsymbol{\tau}_{ext} + \boldsymbol{\tau}_o, \dot{\boldsymbol{q}})$, it would be sufficient to prove that the compliance controller $\boldsymbol{\tau}_o$, with $\boldsymbol{F}_i$ defined according to (3.58), is passive with respect to the port $(-\boldsymbol{\tau}_o, \dot{\boldsymbol{q}})$, to guarantee the stable interaction of the controlled robot with arbitrary passive environments.

As shown earlier, the interconnection between the tank and the task-level controllers is lossless due to the power-preserving Dirac structure (3.64). We can therefore analyze the passivity of this subsystem with a combined storage

$$S_o = \sum_{i=1}^{r} S_{i,c} + E_t. \tag{3.68}$$

The time derivative of S_o is given by

$$\dot{S}_o = \sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{K}_{i,c} \tilde{\tilde{\boldsymbol{x}}}_i + \dot{E}_t \tag{3.69}$$

where using (3.62) and (3.64), $\dot{E}_t$ can be simplified as

$$\dot{E}_t = \sum_{i=1}^{r} \beta_i \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{D}_i \dot{\tilde{\boldsymbol{x}}}_i - \sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{F}_{i,d} - \sum_{i=2}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{F}_{i,p} \tag{3.70}$$

inserting this into (3.69), and expanding further

$$\dot{S}_o = -\sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{F}_i - \sum_{i=1}^{r} (1-\beta_i) \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{D}_i \dot{\tilde{\boldsymbol{x}}}_i. \tag{3.71}$$

Recalling that $\dot{\tilde{\boldsymbol{x}}}_i = \boldsymbol{J}_i \boldsymbol{P}_i^T \dot{\boldsymbol{q}}$, together with (3.52),

$$\dot{S}_o = -\dot{\boldsymbol{q}}^T \boldsymbol{\tau}_o - \sum_{i=1}^{r} (1-\beta_i) \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{D}_i \dot{\tilde{\boldsymbol{x}}}_i \tag{3.72}$$

where according the definition of β_i, the last term represents the dissipated power in the system and is negative definite, which implies that the controller is passive with respect to the port $(-\boldsymbol{\tau}_o, \dot{\boldsymbol{q}})$. It is straightforward now to show the overall passivity of the controlled robot described by (3.50) with the storage

$$S_t = S_o + \frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{M} \dot{\boldsymbol{q}} \tag{3.73}$$

with the rate

$$\begin{aligned} \dot{S}_t &= \dot{S}_o + \dot{\boldsymbol{q}}^T \boldsymbol{M} \ddot{\boldsymbol{q}} \\ &= -\dot{\boldsymbol{q}}^T \boldsymbol{\tau}_o - \sum_{i=1}^{r} (1-\beta_i) \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{D}_i \dot{\tilde{\boldsymbol{x}}}_i + \dot{\boldsymbol{q}}^T \boldsymbol{\tau}_o + \dot{\boldsymbol{q}}^T \boldsymbol{\tau}_{ext} \\ &= \dot{\boldsymbol{q}}^T \boldsymbol{\tau}_{ext} - \sum_{i=1}^{r} (1-\beta_i) \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{D}_i \dot{\tilde{\boldsymbol{x}}}_i \end{aligned}$$

where the well known skew symmetry of $\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ was used. This proves passivity with respect to the port $(\boldsymbol{\tau}_{ext}, \dot{\boldsymbol{q}})$ through which the robot interacts with the external environment, hence guaranteeing a stable interaction.

3.4 Safety-Aware Energy Tanks

The introduction of the energy tanks guarantees one aspect of safety by putting an upper limit on the potentially destabilizing behavior in the system. While this ensures a stable interaction by preserving passivity, absolute safety cannot be ensured since a degree of activity is still allowed in the system. Depending on the energy budget assigned through the initial tank energy E_0 and the tasks definition, dangerous behaviors can still occur. For instance, a sharply increasing stiffness profile will lead to a large increase in the robot kinetic energy, which would result in large impact forces and a higher risk of injuries or robot damage in case of potential collisions. In the view of the author, this poses a major shortcoming in the energy tank literature, where the design of the tank is often not related to a clear safety metric.

3.4.1 Choice of initial Tank energy

One way to achieve a tank design that respects kinetic energy constraints is through the choice of the initial tank energy. The expression for the rate of kinetic energy during free motion (i.e. $\boldsymbol{\tau}_{ext} = \mathbf{0}$) for a gravity-compensated robot is equal to the injected power through its port of interaction $(\dot{\boldsymbol{q}}, \boldsymbol{\tau})$. This can be expressed as

$$\begin{aligned} \dot{E}_{ke} &= \dot{\boldsymbol{q}}^T \boldsymbol{\tau} = \sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{F}_i \\ &= \sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{F}_{i,c} + \sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{F}_{i,d} + \sum_{i=2}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{F}_{i,p} \quad , \end{aligned} \tag{3.74}$$

where $\bar{\boldsymbol{x}}_i^T = \dot{\boldsymbol{q}}^T \boldsymbol{P}_i \boldsymbol{J}_i^T$ according to (3.57), and where the expression for $\boldsymbol{F}_i$ in (3.58) was used. Recalling the expression of the rate of tank energy (3.70), and assuming for simplicity that $\beta_{1..r} = 0$ which means that refilling of the tank through dissipation is not allowed, (3.74) can be further expanded as

$$\dot{E}_{ke} = -\sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{K}_{i,c} \tilde{\tilde{\boldsymbol{x}}} - \sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{D}_i \dot{\tilde{\boldsymbol{x}}}_i - \dot{E}_t. \tag{3.75}$$

At a given time instant, the kinetic energy $E_{ke,t}$ can be expressed as

$$\begin{aligned} E_{ke,t} = E_{ke,o} &- \int_0^t \sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{K}_{i,c} \tilde{\tilde{\boldsymbol{x}}} - \int_0^t \sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{D}_i \dot{\tilde{\boldsymbol{x}}}_i \\ &+ E_0 - E_t \end{aligned} \tag{3.76}$$

and assuming that the initial robot kinetic energy is zero

$$\begin{aligned} E_{ke,t} = \sum_{i=1}^{r} (\frac{1}{2} \tilde{\tilde{\boldsymbol{x}}}_{i,0}^T \boldsymbol{K}_{i,c} \tilde{\tilde{\boldsymbol{x}}}_{i,0} - \frac{1}{2} \tilde{\tilde{\boldsymbol{x}}}_{i,t}^T \boldsymbol{K}_{i,c} \tilde{\tilde{\boldsymbol{x}}}_{i,t}) - \\ \int_0^t \sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T \boldsymbol{D}_i \dot{\tilde{\boldsymbol{x}}}_i + E_0 - E_t \end{aligned} \tag{3.77}$$

which means that at any given time, the robot kinetic energy is the sum of the total potential energy supplied by all the constant spring parts, the dissipated energy through the dampers and the energy supplied from the tank. The objective now is to ensure that the robot kinetic energy remains always below a safe maximum $E_{ke,max}$. In (3.77), since the constant spring potential $\frac{1}{2} \tilde{\tilde{\boldsymbol{x}}}_{i,t}^T \boldsymbol{K}_{i,c} \tilde{\tilde{\boldsymbol{x}}}_{i,t}$ describes a passive system, the sign of the dissipated power is negative definite and the energy tank is bounded by E_0, the maximum value (3.77) can attain is $\sum_{i=1}^{r} \frac{1}{2} \tilde{\tilde{\boldsymbol{x}}}_{i,0}^T \boldsymbol{K}_{i,c} \tilde{\tilde{\boldsymbol{x}}}_{i,0} + E_0$. Therefore, to satisfy kinetic energy constraints, a suitable choice for E_0 can be made as

$$E_0 = E_{ke,max} - \sum_{i=1}^{r} \frac{1}{2} \tilde{\tilde{\boldsymbol{x}}}_{i,0}^T \boldsymbol{K}_{i,c} \tilde{\tilde{\boldsymbol{x}}}_{i,0} \tag{3.78}$$

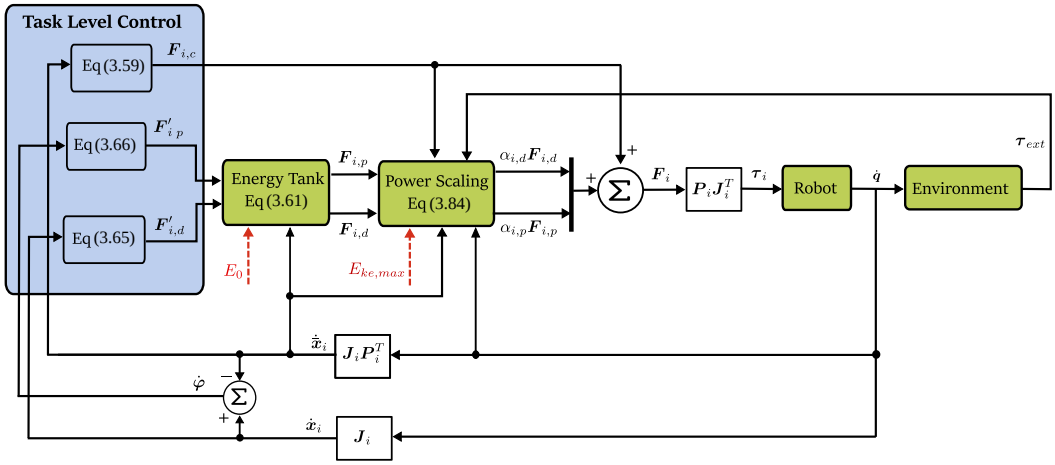


Fig. 3.7: Block Diagram detailing the flow of signals in our system. The red dotted lines entering the blocks "Energy Tank" and "Power Scaling" indicate user-defined parameters, namely the initial tank energy and the maximum kinetic energy allowed in the system. ©2022 IEEE [17].

It is important to note here that we are able to explicitly decouple the contributions to the kinetic energy of the energy extracted from the tank, from the other passive elements of the control action described by the energy injected through the port $(\dot{\tilde{\boldsymbol{x}}}_i^T, \boldsymbol{F}_{i,c})$. This decoupling is made transparent due to the particular choice of tank causality based on unified impedance, thanks to the superposition principle of impedances [231]. With reference to the bond graph in figure 3.6, this can be highlighted by the 1-Junction before the Modulated transformer right before the robot, where all the nodes share the same flow variable $\dot{\tilde{\boldsymbol{x}}}_i$, which is set by the robot admittance (i.e. inertia), as the modulated transformer $\boldsymbol{P}^T\boldsymbol{J}_i$ acts dually both in the effort and flow paths.

3.4.2 Power Flow Regulation

While the choice of a suitable E_0 ensures respecting the kinetic energy limitations in the system, as will be shown in the simulations of section 3.5, this choice can be overly conservative and therefore can severely limit the performance. This is due to the fact that the kinetic energy of the robot changes as a function of its real time state, which implies that enforcing the kinetic energy constraints should happen in real time based. This is achieved by using the concept of power flow regulation [164, 165], where the power injected from the tank into the system is monitored and regulated in real time through additional valves, only when needed to respect the constraints. As shown in [165], the authors argued that for a force control task, in addition to the total energy transferred from the tank to the system, managing the rate of this energy is also critical. We follow similar lines of thoughts and adapt this concept to our case where it is desired to respect maximum kinetic energy constraints in the system.

A first-order Euler approximation of the kinetic energy in discrete time can be written as

$$\begin{aligned} E_{ke}(k+1) = E_{ke}(k) + \Delta t(\sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i^T(k)\boldsymbol{F}_{i,c}(k) + \dot{\boldsymbol{q}}(k)^T\boldsymbol{\tau}_{ext}(k)) \\ +\Delta t(\sum_{i=1}^{r} P_{i,d}(k) + \sum_{i=1}^{r} P_{i,p}(k)) \end{aligned} \tag{3.79}$$

where k represents the time step, Δt is the sampling time, $P_{i,d} = \dot{\tilde{\boldsymbol{x}}}_i^T\boldsymbol{F}_{i,d}$ and $P_{i,p} = \dot{\tilde{\boldsymbol{x}}}_i^T\boldsymbol{F}_{i,p}$. The last two terms indicate the power transferred from the tank to the controllers over the sampling period, which describe the potentially non-passive control behaviors we aim to regulate. To this purpose, at each time instant, if the computed $E_{ke}(k+1) > E_{ke,max}$, we compute the maximum power demand for the tank

$$\begin{aligned} P_{max}(k) = \frac{1}{\Delta t}(E_{ke,max} - E_{ke}(k)) - \Delta t\sum_{i=1}^{r} \dot{\tilde{\boldsymbol{x}}}_i(k)^T\boldsymbol{F}_{i,c}(k) - \\ \Delta t\dot{\boldsymbol{q}}(k)^T\boldsymbol{\tau}_{ext}(k) - \Delta t(\sum_{i=1}^{r} \check{P}_{i,d}(k) + \sum_{i=1}^{r} \check{P}_{i,p}(k)) \end{aligned} \tag{3.80}$$

which represents the allowed power that all the passivity violating ports can inject into the system. The terms $\check{P}_{i,d}(k)$ and $\check{P}_{i,d}(k)$ are defined as

$$\begin{aligned} \check{P}_{i,d}(k) &= \begin{cases} P_{i,d}(k) & \text{if } P_{i,d}(k) \leq 0 \ , \\ 0 & \text{otherwise}\,. \end{cases} \\ \check{P}_{i,p}(k) &= \begin{cases} P_{i,p}(k) & \text{if } P_{i,p}(k) \leq 0 \ , \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{3.81}$$

$\forall i = 1...r$. With the addition of the last two terms in equation (3.80), we are enforcing only the regulation of those passivity violating ports with a transmitted power greater than zero. Having computed $P_{max}(k)$, we now aim to divide this power among the individual power ports in order to regulate the control law for each task separately. One possibility is to divide the power among all ports equally. A smarter approach however would take into account the task priorities, where a high priority task should be less affected than a lower priority one. We propose to ensure this by solving the following optimization problem in real time

$$\max_{\alpha_{i,d}(k),\alpha_{i,p}(k)} \quad \sum_{i=1}^{r} \alpha_{i,d}(k)\hat{P}_{i,d}(k) + \sum_{i=2}^{r} \alpha_{i,p}\hat{P}_{i,p}(k) \tag{3.82a}$$

$$\text{subject to} \quad 0 \leq \alpha_{i,p}(k), \alpha_{i,d}(k) \leq 1, \tag{3.82b}$$

$$\sum_{i=1}^{r} \alpha_{i,d}(k)\hat{P}_{i,d}(k) + \sum_{i=2}^{r} \alpha_{i,p}(k)\hat{P}_{i,p}(k) \leq P_{max}(k), \tag{3.82c}$$

$$\alpha_{i,d}(k) \geq \alpha_{j,d}(k), \alpha_{i,p}(k) \geq \alpha_{j,p}(k) \ \ \forall i > j, \tag{3.82d}$$

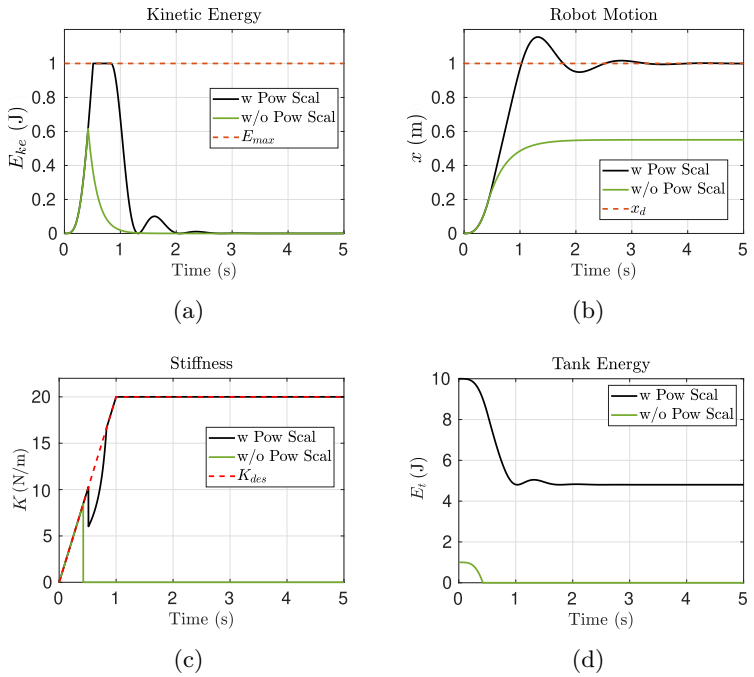


Fig. 3.8: Motivation of the power scaling approach. Results of a 1-DOF robot simulation showing the robot kinetic energy, robot motion, the desired (dotted) and actual stiffness profile (solid) and the tank energy. The results based on the initial assignement of the tank energy and without power scaling are shown in green (w/o Pow Scaling), while the results with the power scaling activated are shown in black (w Pow Scaling). ©2022 IEEE [17].

where $\forall i = 1...r$,

$$\hat{P}_{i,d}(k) = \begin{cases} P_{i,d}(k) & \text{if } P_{i,d}(k) > 0 \ , \\ 0 & \text{otherwise} \, . \end{cases}$$
$$\hat{P}_{i,p}(k) = \begin{cases} P_{i,p}(k) & \text{if } P_{i,p}(k) > 0 \ , \\ 0 & \text{otherwise} \, . \end{cases} \tag{3.83}$$

which ensures maximizing the power that can be extracted from the tank, while enforcing the constraint that the maximum power extractable from the tank is always less than the maximum allowable demand P_{max}. Furthermore, with (3.82d), we ensure that a low priority task will be scaled more than a higher priority ones. It is important to note here that the values $\hat{P}_{i,d}(k)$, $\hat{P}_{i,p}(k)$ and $P_{max}(k)$ are computed prior to the optimization and are assumed constant over the sampling period. This means that the cost function and the constraints are linear in the optimization variables, and in consequence renders our optimization problem a convex linear program with a constrained solution set [232], which could be easily solved in real time.
In order to actually implement the power flow limitation scheme, the Dirac stricture

(3.64) is modified according to:

$$\begin{bmatrix} \boldsymbol{F}_{i,d} \\ \boldsymbol{F}_{i,p} \\ u_{t,i} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \frac{\alpha_{i,d}\sigma_i \boldsymbol{F}'_{i,d}}{x_t} \\ \mathbf{0} & \mathbf{0} & \frac{\alpha_{i,p}\sigma_i \boldsymbol{F}'_{i,p}}{x_t} \\ -\frac{(\alpha_{i,d}\sigma_i \boldsymbol{F}'_{i,d})^T}{x_t} & -\frac{(\alpha_{i,p}\sigma_i \boldsymbol{F}'_{i,p})^T}{x_t} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_i \\ \dot{\boldsymbol{x}}_i \\ y_{t,i} \end{bmatrix}, \tag{3.84}$$

which means that the control force for each task controller sent to the robot is computed as

$$\boldsymbol{F}_i = \boldsymbol{F}_{i,c} + \alpha_{i,d}\sigma_i \boldsymbol{F}'_{i,d} + \alpha_{i,p}\sigma_i \boldsymbol{F}'_{i,p}. \tag{3.85}$$

The detailed controller implementation and the sequence of computations in our system is depicted in the form of a conventional signal-based block diagram in Figure 3.7, which highlights the input-output causality of the different system components.

3.5 Experimental Validation

In order to validate our approach, we present in the following a series of experiments both in simulation and real robot hardware (KUKA LWR 4). We first motivate the power scaling approach as compared to the choice of the initial tank energy in order to limit the maximum kinetic energy in the system. We then validate the time-varying hierarchical compliance controller implementation in a more complex task setting where three hierarchies are assigned. Then, we move to real robot validations where using the proposed controller we achieve a button pressing task. Finally, we test the proposed power flow regulation scheme on the robot when subject to systematic environmental disturbances and in a human-robot interaction scenario. For the simulations, they were implemented on Matlab/Simulink on a standard core i7 desktop PC with 16GB of RAM, and a sampling frequency of 1000 Hz was used during the simulations. As for the real robot experiments, they were implemented on a PC with similar specs. In order to control the robot, we used the Fast Research Interface (FRI) library implemented in C++ to establish communication between the PC and the robot low level controller. The control frequency for robot experiments was 500 Hz.

Motivation of the Power Scaling approach

In this subsection, we aim to highlight the benefits in performance of the power scaling approach as compared to assigning the initial tank energy. To this end, we conducted a simulation on a simple 1-DOF robot with a unit mass, driven by a spring with a linearly increasing time-varying stiffness profile. The robot dynamics

can be described as

$$\ddot{x} = -\sigma_d \underbrace{k(t)(x - x_d)}_{F_d} - D\dot{x} \ , \ k(t) = \begin{cases} 20, & \text{if } k(t) > 20, \\ 20t & \text{otherwise} \end{cases} \tag{3.86}$$

and with an energy tank interconnected to it in order to compensate for the non-passive varying stiffness behavior. The tank dynamics are assigned as

$$\dot{E}_t = -\sigma_d \dot{x} F_d \ , \ \sigma_d = \begin{cases} 1, & \text{if } E_t > 0, \\ 0 & \text{otherwise} \end{cases} \tag{3.87}$$

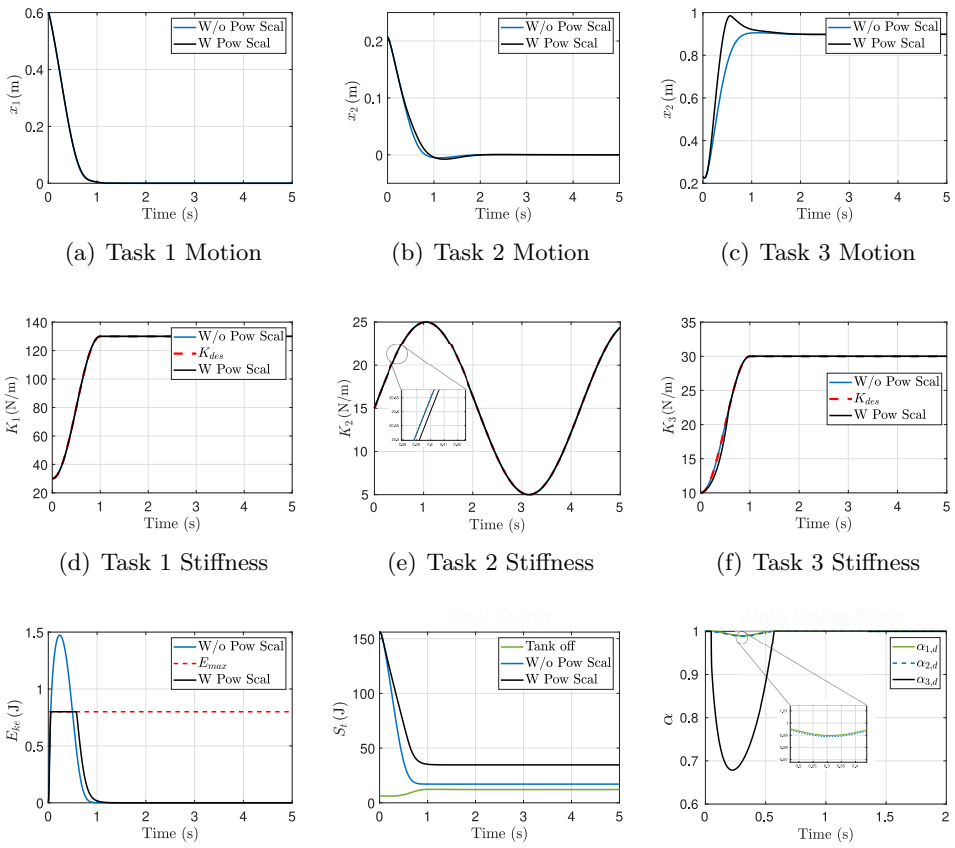
which ensures a passive system with the storage function $\frac{1}{2}x^2 + E_t$. The goal is to keep the kinetic energy less than the allowed kinetic energy limit $E_{ke,max}$ set to 1 J. This can be achieved using two ways

- Setting the initial tank energy $E_{t,0} = E_{ke,max}$
- The Power Scaling approach, where an arbitrary initial tank energy is assigned

In order to minimize the effect of dissipation, the damping was set to a relatively low value $D = 1N.m^2$.
The results of this simulation are depicted in Figure 3.8 which shows the total kinetic energy in the system, the actual robot motion, the actual and desired stiffness profiles and the tank energy. Clearly, solely assigning the initial tank energy maintains the kinetic energy below 1 J, however the method is overly conservative as the kinetic energy only attains a maximum of 0.6 J failing to exploit the maximum possible performance. This also severely affects the performance since as soon as the tank becomes depleted, the robot fails to track the desired varying stiffness profile and stops prematurely, in order to preserve passivity in the system. Whereas for the power regulation method, the robot is able to reach the desired set point and maintain the kinetic energy below the desired limit, at the expense of temporarily sacrificing the tracking of the stiffness profile.

3.5.1 Simulation Study

In this section, we aim to validate our theoretical findings with a more complex task hierarchy with three priority levels. To this end, we performed a simulation study on the 4-DOF planar manipulator described in section 3.2.4. For the primary task, the goal is to achieve an X/Y cartesian compliance at the end-effector with a time-varying stiffness, while regulating its position to a desired setpoint. Similarly, the secondary task aims at regulating the cartesian position of the second joint along the y axis, with a sinosuidal time-varying stiffness profile. Finally, the lowest level in the hierarchy is a joint level impedance control, with a desired configuration chosen deliberately to be non-feasible given the constraints imposed by the higher level tasks. For both the primary and third level tasks, we use the smoothly rising function $\mathcal{K}(t, k_f, t_{min}, t_{max})$ in (3.45) to encode the stiffness profile. We compare the following scenarios



(a) Task 1 Motion (b) Task 2 Motion (c) Task 3 Motion

(d) Task 1 Stiffness (e) Task 2 Stiffness (f) Task 3 Stiffness

(g) Kinetic Energy (h) Total Storage (i) Alpha Scaling Factor

Fig. 3.9: Simulation Results on 4 DOF robot showing the timely evolution of the task coordinates on all three levels, the stiffness profiles, as well as the kinetic energy, total energy and the power scaling factors. We compare the case where the energy tank is switched off (Tank off), the tank is switched on but without power scaling (W/o Pow Scal), and finally, the tank on with the power scaling activated (W Pow Scal). ©2022 IEEE [17].

- Case 1: Tank Off
- Case 2: Tank On, without kinetic energy constraints (w/o Pow Scal)
- Case 3: Tank On, with a maximum kinetic energy 0.8 J using the power regulation approach (w Pow Scal)

The results of this study are shown in Figure 3.9, where depicted are the motion on all the levels, the stiffness, the kinetic energy and the total energy as well as the scaling factors for the time-varying stiffness control action of all three tasks (i.e $\alpha_{i,d}$). For the sake of clarity, we show only the results of the total system energy for Case 1, since otherwise the plots are almost identical to Case 2. For Case 1, the system is clearly not passive since the total energy of the system temporarily

increases. When the tank is introduced in Case 2, the passivity violating behavior is monitored and compensated by the tank, which can be confirmed as the energy monotonically decreases. Note that, the steady state total energy for Case 1 is lower compared to Case 2 and 3. This is due to the fact that the expression for the total energy for Cases 2 and 3 contains also the virtual tank energy (equation (3.73)), and therefore the instantaneous energy in the system for these cases is affected by the choice of initial tank energy. Nevertheless, physically speaking, the kinetic energy for Cases 1 and 2 is almost the same, and is even lower for Case 3, wherein the use of power scaling enforces further additional safety constraints in the system keeping the kinetic energy below the desired limit. This comes at the expense of scaling down temporarily the desired stiffness profile. Thanks to the use of the optimization that takes the task priorities into account, the stiffness remains nearly unaffected for the first two priority levels, and gets slightly distorted for the lowest priority level. Similarly on the motion level, the performance on last level is affected the most due to the conflict with the higher priority tasks. In consequence the task error increases, as compared to the primary and secondary tasks, where the error is regulated and reaches 0 eventually.

3.5.2 Robot Validation

Button pressing task

In this subsection, we validate our passivity results on a 7 DOF KUKA LWR robot in a task that involves both free motion and interaction with the environment. More specifically, the robot has to reach for an emergency button switch from an initial position with a desired path and achieve the pressing operation. To maximize safety, the posture behavior of the null space is chosen to minimize the so-called reflected mass of the robot [233], where the approach described in [234] is chosen to find the optimal configuration. Therefore, we chose a task hierarchy consisting of three priority levels:

- Cartesian compliance for the end effector position ($\mathbb{R}^3$)
- Orientation compliance holding the initial orientation ($\mathbb{R}^3$)
- Joint-level compliance for the minimization of the robot reflected mass ($\mathbb{R}^7$)

For the lowest priority task, the desired configuration corresponds to the local minimum of the reflected mass along the direction of negative gradient, and was found by iteratively integrating $\boldsymbol{Y}_2$ which spans the null space of the stacked Jacobians of the first two tasks, starting from the initial robot configuration [234]. Due to the large initial error between the initial and desired posture, we used (3.45) to generate a smoothly rising stiffness profile.

For the primary task, we encode both the interaction behavior and the motion profile in our stiffness potential, thereby, avoiding the need of a time indexed trajectory that would require a tracking controller[3]. More specifically, we use a single user

[3]Our derived analysis holds only for the regulation case

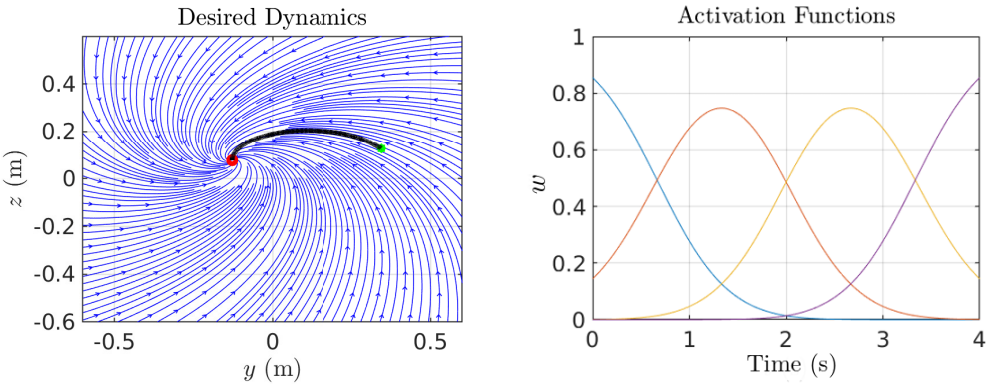


Fig. 3.10: Learnt Dynamics for the primary task, and the time-varying activation functions associated with the linear springs. ©2022 IEEE [17].

demonstration to encode a non-linear motion profile, in a spring potential composed of the weighted sum of linear springs, activated by a time-varying activation function. This is achieved by solving the following optimization problem

$$\min_{K_{1,l..L}} \quad \sum_{t=0}^{T} || \sum_{l=1}^{L} w_l(t) \boldsymbol{K}_{1,l}(\boldsymbol{x}_{1,g} - \boldsymbol{x}_{1,t}) - \dot{\boldsymbol{x}}_{1,t} || \tag{3.88a}$$

$$\text{subject to} \quad \boldsymbol{K}_{1,l} + \boldsymbol{K}_{1,l}^T \succ \boldsymbol{0}, \tag{3.88b}$$

where T is the total demonstration time, $\boldsymbol{x}_{1,t}$ is the actual position of the end-effector during the demonstration at a given time instant, and $\boldsymbol{x}_{1,g}$ is the final goal point of the demonstration. The number of linear springs is denoted by L, while $w_l(t)$ is a time-dependent radial basis activation function of the l-th linear spring. The constraint (3.88b) has been added to ensure linear spring terms with positive definite stiffness. Figure 3.10 show the used activation function and the phase plane of the learned spring potential plotted over motion directions y and z. As the phase plane show, solutions converge to the attractor $\boldsymbol{x}_{1,g}$ regardless of the initial position. Finally, in order to provide the needed force along the desired dynamics, we devise a controller structure inspired by [131], that the authors proposed to track the motion generated by a first-order dynamical system. Additionally, we use the activation function (3.45), leading to the following expression of time-varying stiffness for the primary task

$$\boldsymbol{K}_{1,d}(t) = \mathcal{K}(t, k_f, t_{min}, t_{max}) \sum_{l=1}^{L} w_l(t) \boldsymbol{K}_{1,l} \tag{3.89}$$

and the corresponding spring force $\boldsymbol{F}'_{1,d} = -\boldsymbol{K}_{1,d}(t)(\boldsymbol{x}_1 - \boldsymbol{x}_{1,g})$. The parameters k_f, t_{min}, t_{max} were tuned to achieve a smooth onset of the motion, and rising gradually to generate a high stiffness when approaching the button to generate the necessary pushing force.

The results of this experiment are shown in Figure 3.11 where we show the actual robot motion, the robot reflected mass, the rate of the total storage function during

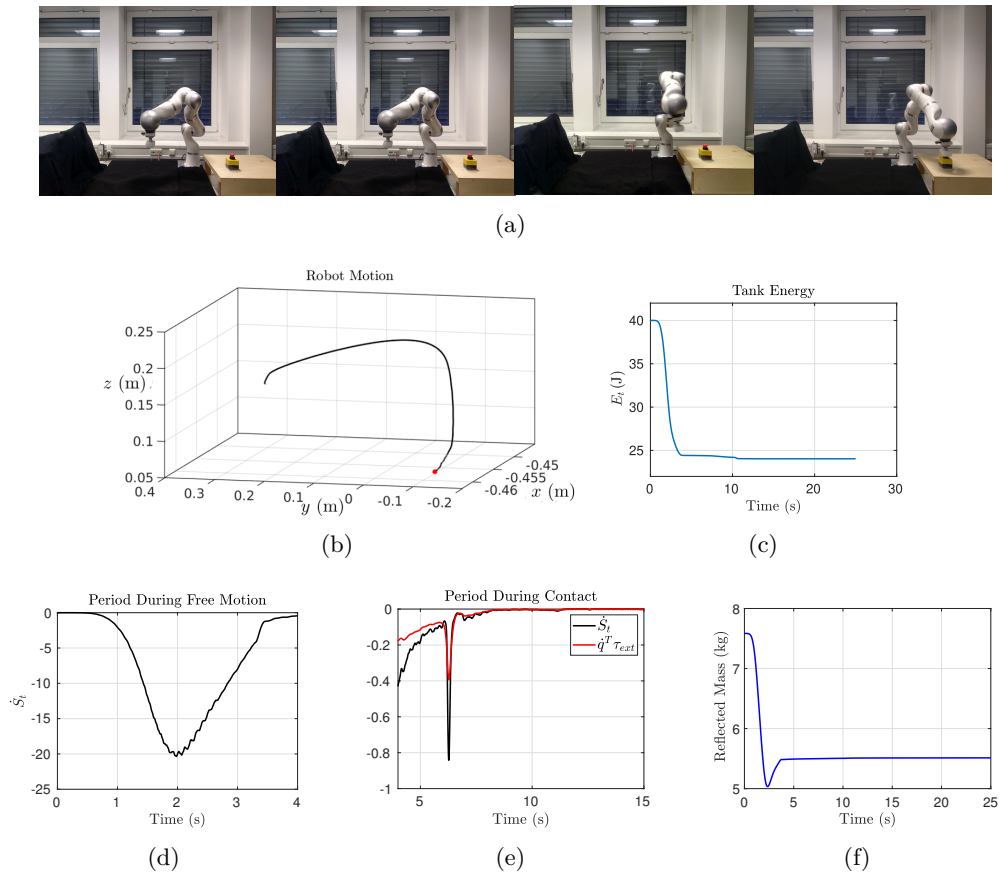


Fig. 3.11: Button Pressing Experimental Results. Top row shows snapshots of the robot experiment starting from the initial configuration, and ending with the final robot state after pressing the button. The second row shows the actual cartesian motion in 3D, while the last row shows the rate of total energy during free motion, during contact as well as the power supplied from the environment, and the robot reflected mass. ©2022 IEEE [17].

free motion as well during the pressing phase. Clearly, our controller is able to achieve the task objectives, demonstrated by the ability of the end effector to reach the desired goal configuration and the minimization of the robot reflected mass thanks to the null space control action. Furthermore, our system remains passive throughout the task, since during free motion $\dot{S}_t$ remains below 0, while during contact the total energy in the system is always less than or equal the supplied energy by the environment i.e. $\dot{S}_t \leq \dot{\boldsymbol{q}}^T \boldsymbol{\tau}_{ext}$.

Robustness to disturbances

While the previous results showed the ability of our controller to perform stable task execution, in this subsection, our aim is to demonstrate in more depth the benefit of augmenting the tank with additional safety constraints, by highlighting its robustness to environmental disturbances. To this end, we chose a task hierarchy consisting of

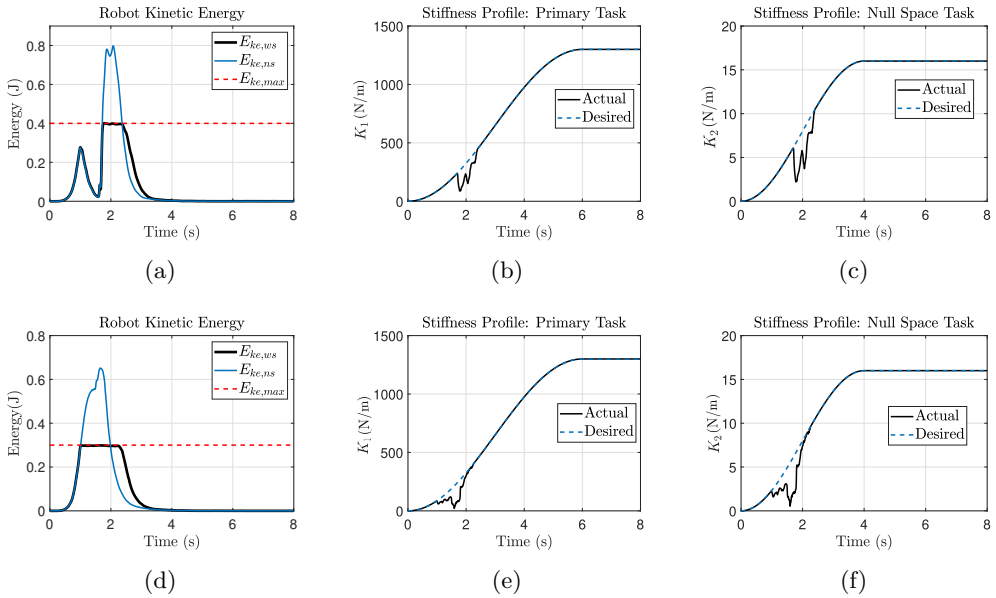


Fig. 3.12: Experimental Results when robot is subjected to disturbances. The first row corresponds to the disturbance applied by simulating an artificial obstacle that blocks the robot motion, while the results of applying a force disturbance are depicted in the second row. For both disturbance types, we show the robot kinetic energy when the power scaling is activated ($E_{ke,ws}$) and when it is not ($E_{ke,ns}$), in addition to the maximum allowed kinetic energy. In the second and third columns, we show the desired and actual (scaled) stiffness profiles for both the primary and null space tasks. ©2022 IEEE [17].

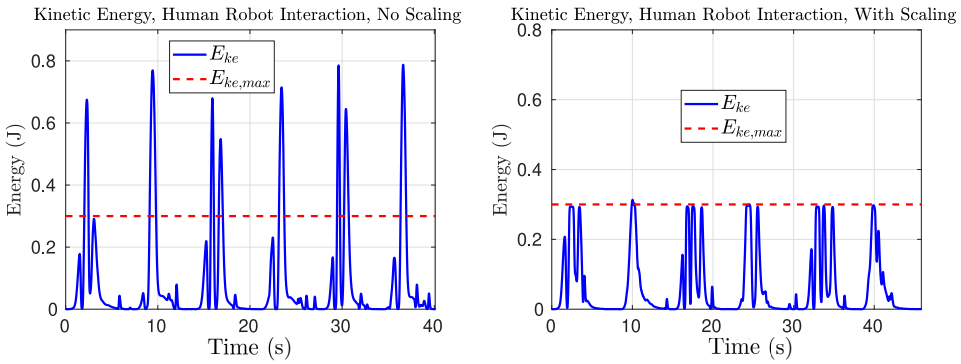


Fig. 3.13: Robot Kinetic results from the Human-Robot interaction experiment. Left figure show the results without applying power scaling, while in the right one scaling is applied. ©2022 IEEE [17].

two tasks, where the first task ($\mathbb{R}^6$) is to regulate the end-effector position along the y direction to a desired final position, while maintaining the initial position and orientation in the other Cartesian directions. Similarly, the null space task was a joint-level impedance ($\mathbb{R}^7$) to reach a desired final configuration without disturbing the end-effector task. For both tasks, we used the same smooth activation (3.45) to

encode the stiffness profile. During the tasks execution, the robot was subject to two types of disturbances:

- A Force of 10 N applied at the end effector along the direction of motion for 0.2 s between $t = 1s$ and $t = 1.2s$, with $E_{ke,max}$ =0.3 J
- Disabling the control commands (i.e. $\boldsymbol{\tau}_o = \mathbf{0}$) between $t = 1s$ and $t = 1.4s$, without stopping the time counter for the stiffness profiles, with $E_{ke,max}$ =0.4 J

In the first case, the disturbance force is intended to cause a sudden acceleration which would lead to the violation of the kinetic energy constraints. On the other hand, the second case represents a common problem for any robot driven by an open-loop simulated time-indexed trajectory. That is, the trajectory generator does not have any feedback on the actual robot state. This becomes problematic if for instance the robot gets stuck behind an obstacle, leading to a large jump in acceleration and in consequence the robot kinetic energy when the obstacle is suddenly released. The goal of the second case is to showcase such a situation, however in a more controlled setting where the control commands are computed, but not sent to the robot actuators during the freezing period. For the two cases, we compare the performance of the controller with the power scaling where the kinetic energy constraints are active, and inactive. The kinetic energy results for this experiment, as well as the actual scaled (i.e $\alpha_{i,d}\boldsymbol{K}_{i,d}(t)$) and desired ($\boldsymbol{K}_{i,d}(t)$) stiffness profiles for both the primary and null space task when the power scaling is activated, are shown in Fig. 3.12. For the first disturbance case (Fig. 3.12(d)), the robot's kinetic energy clearly remains below the safe limit with the power scaling active, as compared to without scaling where the constraints are violated. This is also the case for the second disturbance (Fig. 3.12(a)), where following the period the robot motion is artificially stopped ($t \approx 1s$), the robot's kinetic energy experiences a sudden jump exceeding the safe limit when the power scaling is not activated, as compared to the controller with the power scaling active where the energy constraints are always respected. For both disturbance cases, enforcing kinetic energy constraints results in temporarily sacrificing the tracking of the desired stiffness profiles, for both the primary (Fig. 3.12(e), Fig. 3.12(b)) and null space tasks (Fig. 3.12(f), Fig. 3.12(c))

Human-Robot Interaction

In the final part of the validation, we test the robustness of our approach in a more randomized setting, where a human physically interacts with the robot. The desired hierarchy for the primary and null space tasks was set similar to the previous subsection, however here, the robot should alternate between two desired set-points on both the primary and null space task levels, in order to achieve a repetitive motion. While the robot is moving, the human applies several random disturbances by pushing the robot at various locations, and attempting also to temporarily block the robot motion. The experiment is conducted twice, with and without the activation of the power scaling scheme, and where the goal was to keep the total kinetic energy of the robot below $E_{ke,max}$ =0.3 J. As shown in Fig. 3.13 left, the kinetic energy exceeds

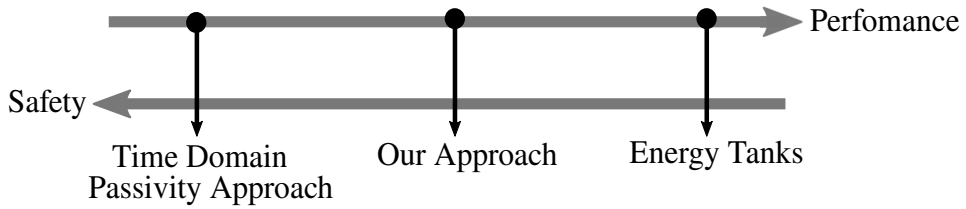


Fig. 3.14: Comparison between TDPA, classical energy tanks and our approach. ©2022 IEEE [17].

the safe limit without applying the scaling, resulting also in a more aggressive robot behavior. On the other hand, when applying the scaling (Fig. 3.13, right), the kinetic energy remains below the assigned limit, and the motions are smoother and less abrupt.

3.6 Discussion

Passivity is an important requirement for guaranteeing a safe behavior for a robot interacting with uncertain or unknown environments, which can be violated by the presence of certain control actions. Unfortunately, these actions can stem from specific performance requirements (e.g. Projection operators or variable impedance behaviors) and therefore, simply omitting them would not be a practical solution. In this regard, energy tanks represent an elegant and a mathematically sound approach to deal with such situations. The tank is simply a passive system interconnected to the controllers in a power-continuous manner, that preserves the execution of these potentially non-passive control actions, while ensuring passivity in the system. This was confirmed in our simulation results of section 3.5.1, where the introduction of the tank results in the system becoming passive. While practically the performance is similar to the non-passive case, the tank can be seen as a monitoring mechanism that keeps track of the passivity violating energy in the system, ensuring that it always remains bounded. In case the energy requested by these control actions exceeds a certain threshold, the tank would be drained of energy resulting in the deactivation of these actions, preserving safety at the expense of some performance.

An important parameter that should be chosen with care is the allocated budget assigned through the choice of the initial tank energy. A very high budget can mask unstable behaviors such as oscillations or large control forces, while on the other hand, a low budget would compromise the performance by deactivating the non-passive control actions prematurely. Therefore, we argue that a suitable guideline for choosing this budget would be based on a concrete safety metric, for instance the maximum kinetic energy allowed in the system. This also requires a clear formulation of how the "virtual" tank energy gets transformed into physical kinetic energy. Unfortunately however, the results of our toy example in section 3.5 have confirmed our hypothesis that solely assigning the initial tank energy, while effective in limiting the kinetic energy, can unnecessarily restrict the performance.

A more efficient solution to satisfy the kinetic energy constraints, is by regulating the power injected by the passivity violating control actions. Indeed, we have shown in section 3.5.1, that we are able to enforce the safety constraints, by only temporarily sacrificing the performance. This provides more freedom in assigning the initial tank energy. In addition to that, as shown in section 3.5.2, the system becomes more robust to possibly unaccounted for environmental disturbances (e.g. Robot gets stuck behind an obstacle), that can trigger dangerous behaviors from the non-passive control actions (e.g. time-varying stiffness behaviors). A potential limitation however arises in situations where the increase in the kinetic energy in the system can be beyond what can be dissipated from the power scaling. This would correspond to the case where $P_{max}(k)$ in (3.80) would become less than zero, and could happen for example in the presence of an external interaction with the environment, where the energy injected via the port $(\dot{\boldsymbol{q}}, \boldsymbol{\tau}_{ext})$ can be very high. Nevertheless, such a case can be easily checked prior to the optimization problem, and set all the power scaling factors to zero instead of solving the optimization problem. Future work might consider in such a case injecting additional dissipation in the system, or augmenting the energy tank with a release valve, as done in [235].

Finally, it is also worth mentioning that our power scaling scheme shares noticeable similarities with the Time-Domain passivity control approach (TDPA) [122, 123], commonly used for guaranteeing the passivity of haptic interfaces [122] and bilateral teleoperators [123]. There, a "passivity observer" uses a discrete-time formulation to measure the energy flow into a two-port network in real-time, whereas the "passivity controller" injects damping to restore the passivity in the system, only when necessary, i.e the network becomes active. Similarly, we exploit a discrete expression for the total kinetic energy in the system at each time instant (eqn. (3.79)), and scale down the passivity violating control actions only when required, i.e the kinetic energy in the system exceeds the safe limit. Also within the TDPA, the work in [236] presented a framework for restoring the passivity of a redundant manipulator whenever its interaction with the environment becomes active. Exploiting redundancy, the authors proposed to distribute the dissipation of energy throughout the decoupled sub-spaces, prioritizing the dissipation in the null space, reducing the injection of damping in the cartesian task space. Our power regulation scheme also prioritizes the preservation of the performance on higher priority tasks, however, instead of dissipation, the non-passive control actions on a lower priority task are scaled more, compared to the higher levels.

In principle, we could have also applied the TDPA also for the preservation of passivity, instead of energy tanks. This however would mean the injection of damping whenever the robot becomes active, which would compromise the performance. It is also well known that the TDPA suffers from practical problems such as the amplification of noise due to the increased damping actions, and insufficient dissipation at low velocities [122]. On the other side, applying the TDPA has the advantage that it is model-free and that the system is never active. Instead, in energy tanks, one deliberately allows for a degree of activity in the system as long as the passivity violating energy does not drain the tank, thereby ensuring that the potentially

non-passive control action and in consequence, the performance, are preserved. Our presented approach combines energy tanks and power flow regulation (Figure 3.14). We use energy tanks to preserve performance while keeping track of the energy dissipated/generated, but complement the tank with an additional safety layer, that also sacrifices performance, however, only temporarily when the energy in the system exceeds a safe limit.

3.7 Summary

In this work, we aimed to further expand the potential of the powerful hierarchical compliance control framework, by incorporating variable stiffness behaviors. The developed controller is capable of realizing a compliance behavior with a variable stiffness, specified by a time-varying profile, for an arbitrary number of tasks, prioritized in a strict hierarchy. Furthermore, we are able to preserve the passivity in the system despite the presence of active control actions resulting from the stiffness variations and the use of null space projections, thanks to the use of energy tanks that monitor and bound the passivity violating energy in the system. In addition to guaranteeing a stable behavior, the safety of our system is further enhanced. This is achieved by ensuring that the maximum energy injected by the tank does not lead to an increase in the kinetic energy of the robot, beyond a safe pre-defined threshold. We first investigated how the initial energy in the tank can be chosen to realize this objective. Then, we proposed a more efficient solution based on limiting the rate of non-passive energy provided by the tank, through power flow regulation. Finally, we validated our approach in simulations and on a real robot, highlighting the ability of our controller to perform stable task execution, as well as its safety and robustness in the presence of unexpected environmental disturbances.

CHAPTER 4

Variable Impedance Teleoperation For Contact Tasks

This Chapter presents an adaptive impedance control architecture for robotic teleoperation of contact tasks featuring continuous interaction with the environment. We use Learning from Demonstration (LfD) as a framework to learn variable stiffness control policies. Then, the learnt state-varying stiffness is used to command the remote manipulator, in order to adapt its interaction with the environment based on the sensed forces. Our system only relies on the on-board torque sensors of a commercial robotic manipulator and it does not require any additional hardware or user input for the estimation of the required stiffness. We also provide a passivity analysis of our system, where the concept of energy tanks is used to guarantee a stable behavior. Finally, the system is evaluated in a representative teleoperated cutting application. Results show that the proposed variable-stiffness approach outperforms two standard constant-stiffness approaches in terms of safety and robot tracking performance. The contents of this chapter are based on the following publication [18]

Y. Michel, R. Rahal, C. Pacchierotti, P. R. Giordano and D. Lee, "Bilateral Teleoperation With Adaptive Impedance Control for Contact Tasks," in IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 5429-5436, 2021.

4.1 Motivation

As reviewed in section 2.1.2, several works have aimed at augmenting teleoperation architectures with VIC, in an attempt to endow teleoperated robots with the human-like physical interaction abilities, that can be acheived with VIC. One way to acheive this objective is through an external hardware interface that attempts to estimate the user arm impedance, as done for example in the teleimpedance approach [16] where the stiffness is based on muscle activity measured via EMG, or using grip-force sensing as in [120] and [118]. On the other hand, works such as [125] allow for

the operator to command the desired stiffness assuming the human has enough knowledge to convey the optimal stiffness for the given task. Intrestingly, in [125], the authors argued that the redundancy in the force control problem should be exploited whenever possible and achieve the task using a low stiffness. This becomes especially relevant in teleoperation, as high stiffness amplifies the so-called induced master motion, which affects safety and makes it difficult to maintain a stable contact with the environment. As analyzed in [237, 238], this induced motion results eventually in violent recoiling of the remote robot during contact, leading to an internal control loop that is unstable under high gains.

The developed approach in this chapter aims to combine the benefits of variable stiffness control with the capabilities of bilateral haptic teleoperation. It endows a robotic teleoperation system with an additional input channel conveying an automatically-estimated desired user stiffness, in addition to the desired motion. Prior works in this topic require additional external hardware, such as grip force sensing or EMG measurements. Furthermore, in works such as [125], additional cognitive load is imposed on the users as, in addition to controlling the position, one also needs to manually control the stiffness. In contrast to previous works, we instead propose a variable stiffness teleoperation architecture that only uses the torque sensors embedded in commercial robotic manipulators, not requiring any additional hardware. In particular, we use user demonstrations to learn a variable stiffness control policy, envisioned for contact tasks that require the user to maintain a continuous interaction with the environment, e.g., drilling, cutting. Inspired by [7], we argue that, for such tasks, sensed external forces act primarily as a disturbance (e.g. friction), but can also convey valuable information for adapting the robot stiffness. We learn the stiffness policies, relying on the stiffness information derived from the task dynamics, together with Gaussian mixture models (GMM) to encode a task model, and Gaussian mixture regression (GMR) for real-time execution. The learnt policy keeps the manipulator stiffness as low as possible for a safe and compliant interaction, increasing it only when needed to compensate for environmental disturbances. With respect to a constant high-stiffness, we mitigate the effects of induced master motion that might lead to dangerous oscillations in the system. On the other hand, with respect to the constant low-stiffness case, we maintain low tracking errors and completion times.

To the best of our knowledge, incorporating variable impedance learning schemes into bilateral teleoperation has not been considered before in the literature. The contributions of this chapter can be summarized as follows:

- We present a variable impedance learning scheme applied to robotic teleoperation which, unlike previous works, does not require any external hardware (such as grip force or EMG sensors), or any cognitive effort from the side of the user in choosing the optimal stiffness.
- We ensure our system is passive by augmenting it with a passivity layer in the form of a global energy tank [149]. This guarantees additional safety and robustness, during interaction with unknown environments.

Fig. 4.1 shows a block diagram summarizing our approach. We demonstrate the

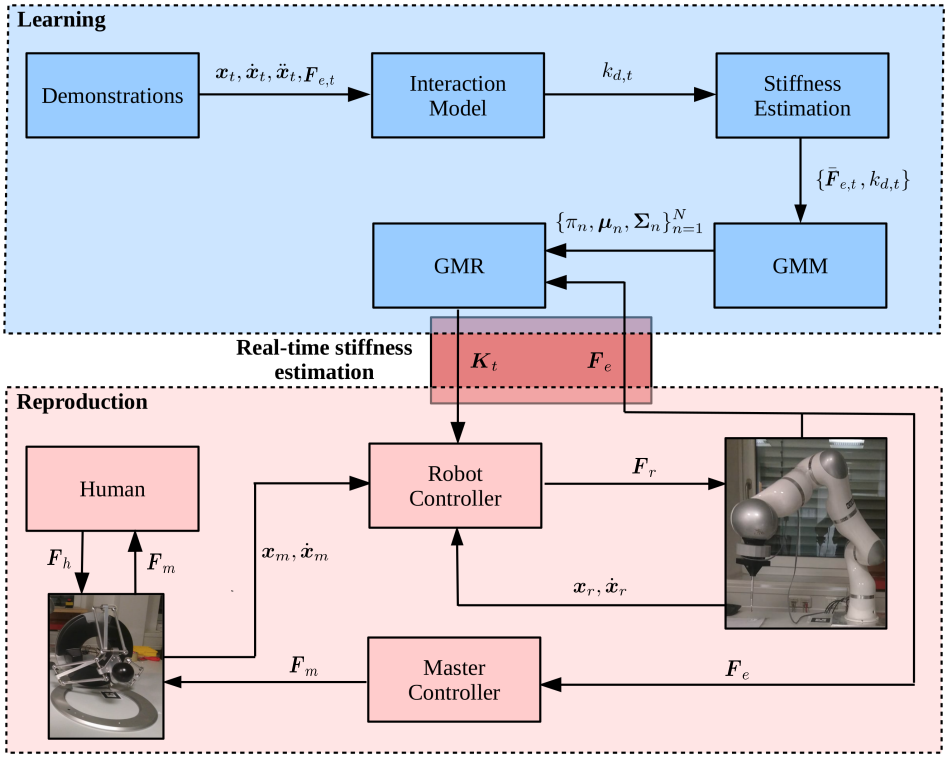


Fig. 4.1: Block diagram summarizing our system. The learning framework allows the dynamic estimation of the desired stiffness $\boldsymbol{K}_t$ for the teleoperation task. This estimation is based on the measured force $\boldsymbol{F}_e$ on the robot. ©2021 IEEE [18].

effectiveness of our method in a representative cutting experiment, similar to [213].

4.2 Task Learning

In this section, we describe our procedure for learning the stiffness profiles. Similar to [7, 79], we use the Mass-Spring-Damper model (MSD) to derive the stiffness. We also propose a method for computing the attractor path of the MSD, consistent with human-motor control theory. Then, we show the GMM/GMR approach and finally describe a method to construct a symmetric positive definite (SPD) stiffness matrix. In the following, for simplicity, we will only consider translational motion tasks. Future work will also consider rotational tasks as well.

4.2.1 Task Dynamics

In order to learn the stiffness from a user demonstrated task, we first need to evaluate the stiffness profile from the corresponding task dynamics. To achieve this, the user provides a set of J demonstrations for Q task situations (a task situation can be for instance a certain material to cut with a specific hardness, in the case of a cutting task). During demonstrations, we collect the robot end-effector positions

$\{\boldsymbol{x}_t\}_{t=1}^T$ and the sensed external forces at the end effector $\{\boldsymbol{F}_{e,t}\}_{t=1}^T$, where t is a time parametrization and T the length of each demonstration. We augment these time series by the velocities $\{\dot{\boldsymbol{x}}_t\}_{t=1}^T$ and accelerations $\{\ddot{\boldsymbol{x}}_t\}_{t=1}^T$ obtained via finite differences. For deriving the stiffness, we assume that along the motion direction, the end-effector behaves as a 1 dimensional unit mass driven by a spring and a damper, subject to the external environment force. This can be expressed as

$$\ddot{x}_t = k_t(x_{t,d} - x_t) - b\dot{x}_t + F_{e,t}, \tag{4.1}$$

where k_t is the time varying stiffness, b is the damping chosen experimentally such that $b > 0$, while $x_{t,d}$ is the unknown spring equilibrium trajectory.
To obtain $x_{t,d}$, we assume that during demonstrations, the human teacher uses a stiffness composed of two parts: 1) an unknown time varying part $k_{d,t}(F_{e,t})$ we seek to derive that continuously adapts to compensate for the external environment force, and 2) a constant part k_c that generates the observed motion dynamics, and such that $k_t = k_{d,t}(F_{e,t}) + k_c$. This assumption is also consistent with the way humans control their end-point stiffness during interaction tasks. As stated in [3], one part of the human arm stiffness is constant and typically low (k_c), ensures stability during free motion, while the other part ($k_{d,t}(F_{e,t})$) increases to compensate for the stiffness of the environment. This implies that (4.1) can be approximated as

$$\ddot{x}_t = k_c(x_{t,d} - x_t) - b\dot{x}_t, \tag{4.2}$$

where k_c is set empirically to a reasonable low value. We can then obtain $x_{t,d}$ as

$$x_{t,d} = k_c^{-1}(\ddot{x}_t + b\dot{x}_t) + x_t. \tag{4.3}$$

Similar to [7], we proceed to define a sliding window L that moves along the demonstrations. We define $\tilde{x}_t = x_{t,d} - x_t$, $y_t = \ddot{x}_t + b\dot{x}_t - F_{e,t}$, and concatenate adjacent data points from all the demonstrations that correspond to the specific task situations defining

$$\begin{aligned} \boldsymbol{X}_t &= [\tilde{x}_{t-L,1} \;\; \tilde{x}_{t-L+1,2} \dots \tilde{x}_{t+L,j}], \\ \boldsymbol{Y}_t &= [y_{t-L,1} \;\; y_{t-L+1,2} \dots y_{t+L,j}]. \end{aligned} \tag{4.4}$$

We then obtain k_t for each time instant via regularized regression as

$$k_t = (\boldsymbol{X}_t^T \boldsymbol{X}_t + \lambda \boldsymbol{I})^\dagger (\boldsymbol{X}_t \boldsymbol{Y}_t^T), \tag{4.5}$$

where λ is a tuning parameter to avoid ill-conditioning and penalize high values of k_t. Having obtained $k_{d,t} = k_t - k_c$, we construct the dataset $\{\bar{f}_{e,1:T,q}, k_{d,1:T,q}\}_{q=1}^Q$ where $\{\bar{f}_{e,1:T,q}\}$ is average sensed forces trajectory, from all demonstrations corresponding to the *jth* task situation.

4.2.2 Stiffness Learning

So far, we are able to derive stiffness profiles from user demonstrations. However, we still need a framework that would allow us to encode and learn a functional

relationship between the sensed external force and the desired stiffness, and use it eventually for real-time reproduction of the learnt skill. To achieve this, we resort to GMM to learn a joint probability distribution between the sensed external force and the desired stiffness. For $\boldsymbol{\xi} = [\xi_i, \xi_o]^T$ referring the data vector, we set $\xi_i = F_{e,t}$ and $\xi_o = k_{d,t}$, and then apply Expectation-Maximization (EM) algorithm to learn the GMM parameters. During reproduction, we use GMR to obtain $k_{d,t}$ as the mean of the distribution of the output conditioned on the input ξ_i, as explained in section 2.2.5 .

4.2.3 Constructing SPD Stiffness

In the aforementioned analysis, we have only considered one cartesian direction. Indeed, we aim to adapt the stiffness along the direction of motion subject to the external environmental disturbance. Nevertheless, we still need to construct a symmetric positive definite stiffness matrix for the robot impedance controller. We achieve this by reconstructing a matrix that provides adaptation selectively along the direction of motion, computed in real-time. To this end, let us define a time window W that contains the measured end-effector velocities $\dot{\boldsymbol{x}}$ over the past W samples up to the current time instant t. We derive the current motion direction as

$$\boldsymbol{u}_t = \frac{\bar{\dot{\boldsymbol{x}}}_t}{||\bar{\dot{\boldsymbol{x}}}_t||} \ , \ \bar{\dot{\boldsymbol{x}}}_t = \frac{1}{W} \sum_{t-W}^{t} \dot{\boldsymbol{x}}_w. \tag{4.6}$$

We also project the current sensed external force along the motion direction

$$F_{e,\boldsymbol{u}_t} = \boldsymbol{F}_e \cdot \boldsymbol{u}_t. \tag{4.7}$$

Let $\boldsymbol{v}_1 = \boldsymbol{u}_t$ and the vectors $\boldsymbol{v}_1 \dots \boldsymbol{v}_n$ span an orthonormal basis of $\mathbb{R}^n$ where $\boldsymbol{v}_2, \dots \boldsymbol{v}_n$ are normalized and orthogonal to $\boldsymbol{v}_1$ and with $n = 3$ as the number of cartesian translational DOF, and the matrix $\boldsymbol{V}(\boldsymbol{x}_t)$ whose columns are the vectors $\boldsymbol{v}_1 \dots \boldsymbol{v}_n$, we compute the stiffness at each time instant as

$$\boldsymbol{K}_{r,t} = \boldsymbol{V}(\boldsymbol{x}_t)\boldsymbol{A}(F_{e,\boldsymbol{u}_t})\boldsymbol{V}(\boldsymbol{x}_t)^T, \tag{4.8}$$

where $\boldsymbol{A}(F_{e,\boldsymbol{u}_t})$ is a diagonal matrix with the first eigenvalue $a_1 = k_c + k_{d,t}(F_{e,\boldsymbol{u}_t})$ that adapts depending on the external force, and with $k_{d,t}(F_{e,\boldsymbol{u}_t})$ computed via the learnt GMM through regression, while the remaining eigen values can be set such that $a_2 \dots a_n \geq 0$.

4.3 Variable Impedance Teleoperation

After learning the GMM model in the previous section, we present here the variable impedance teleoperation system that will adapt the stiffness of the remote robot impedance controller in real-time based on the sensed force from the environment. Let $\boldsymbol{x}_r \in \mathbb{R}^3$ define the position of the remote robot, and $\boldsymbol{x}_m \in \mathbb{R}^3$ define that of the

master device. The master device is modeled as a generic (gravity pre-compensated) mechanical system,

$$\boldsymbol{M}_m(\boldsymbol{x}_m)\ddot{\boldsymbol{x}}_m + \boldsymbol{C}_m(\boldsymbol{x}_m, \dot{\boldsymbol{x}}_m)\dot{\boldsymbol{x}}_m = \boldsymbol{F}_m + \boldsymbol{F}_h, \tag{4.9}$$

where $\boldsymbol{M}_m(\boldsymbol{x}_m) \in \mathbb{R}^{3\times3}$ is the positive-definite and symmetric inertia matrix, $\boldsymbol{C}_m(\boldsymbol{x}_m, \dot{\boldsymbol{x}}_m) \in \mathbb{R}^{3\times3}$ represents Coriolis/centrifugal terms, and $\boldsymbol{F}_m, \boldsymbol{F}_h \in \mathbb{R}^3$ are the control and human forces, respectively. Similarly, on the level of the remote manipulator,

$$\boldsymbol{M}_r(\boldsymbol{x}_r)\ddot{\boldsymbol{x}}_r + \boldsymbol{C}_r(\boldsymbol{x}_r, \dot{\boldsymbol{x}}_r)\dot{\boldsymbol{x}}_r = \boldsymbol{F}_r + \boldsymbol{F}_e, \tag{4.10}$$

where $\boldsymbol{M}(\boldsymbol{x}_r) \in \mathbb{R}^{3\times3}$ is the positive-definite and symmetric inertia matrix, $\boldsymbol{C}_r(\boldsymbol{x}_r, \dot{\boldsymbol{x}}_r) \in \mathbb{R}^{3\times3}$ accounts for Coriolis/centrifugal terms, and $\boldsymbol{F}_r, \boldsymbol{F}_e \in \mathbb{R}^3$ are the control command and external forces exerted on the robot, respectively. For both the master and remote systems, we have the valuable passivity feature that lies in the skew symmetry of $(\dot{\boldsymbol{M}}_i - 2\boldsymbol{C}_i)$ for $i = \{r, m\}$, which implies that the remote robot and master dynamics are passive with respect to the ports $(\boldsymbol{F}_e + \boldsymbol{F}_r, \dot{\boldsymbol{x}}_r)$ and $(\boldsymbol{F}_h + \boldsymbol{F}_m, \dot{\boldsymbol{x}}_m)$, respectively.

The control command on the remote robot is designed as an impedance controller, such that the position of the robot end effector follows that of the master interface

$$\boldsymbol{F}_r = \boldsymbol{K}_{r,t}\left(\boldsymbol{x}_m - \boldsymbol{x}_r\right) - \boldsymbol{D}_r\dot{\boldsymbol{x}}_r, \tag{4.11}$$

where $\boldsymbol{K}_{r,t} \in \mathbb{R}^{3\times3}$ is our learnt state varying stiffness term and $\boldsymbol{D}_r \in \mathbb{R}^{3\times3}$ is a positive definite damping matrix. As for the master controller, it is designed as to reflect the environment force on the remote side as

$$\boldsymbol{F}_m = -\boldsymbol{F}_e - \boldsymbol{D}_m\dot{\boldsymbol{x}}_m, \tag{4.12}$$

where $\boldsymbol{D}_m$ is a positive definite damping matrix, added to improve contact stability.

4.3.1 Passivity Analysis

While the proposed approach benefits from some intrinsic robustness as the stiffness is increased only when needed and as dictated by the interaction with the remote environment, the stable interaction with arbitrary passive environments is still not guaranteed. This is due to the fact the the proposed teleoperator system is non-passive. In general, it is well known that a force-reflecting teleoperation system is non-passive even in the non-delayed case [239]. The problem is further aggravated in case of a varying stiffness, as stiffness variations represent an active action [22,96,240]. Since environments can be assumed passive and the human operator hand impedance is often characterized by a passive velocity to force map [239], it suffices to passify the controlled teleoperator system to guarantee overall stable operation. To achieve that, we will use the concept of energy tanks, where we will augment our controllers with a passivity layer that consists of a global energy tank. The choice of one global tank compared to two local tanks on master and remote robot levels as done in [150]

is motivated by the work in [12], where the authors noted that a global tank is less conservative than local tanks.

First, let us analyze the passivity of our system with the following storage function

$$S = \frac{1}{2}\tilde{\boldsymbol{x}}_r^T \boldsymbol{K}_t \tilde{\boldsymbol{x}}_r + \frac{1}{2}\dot{\boldsymbol{x}}_m^T \boldsymbol{M}_m \dot{\boldsymbol{x}}_m + \frac{1}{2}\dot{\boldsymbol{x}}_r^T \boldsymbol{M}_r \dot{\boldsymbol{x}}_r, \tag{4.13}$$

where $\tilde{\boldsymbol{x}}_r = \boldsymbol{x}_m - \boldsymbol{x}_r$. The derivative of (4.13) along the closed loop dynamics of the system can be expressed as

$$\begin{aligned} \dot{S} = \dot{\boldsymbol{x}}_r^T \boldsymbol{F}_e + \dot{\boldsymbol{x}}_m^T \boldsymbol{F}_h - \dot{\boldsymbol{x}}_r^T \boldsymbol{D}_r \dot{\boldsymbol{x}}_r - \dot{\boldsymbol{x}}_m^T \boldsymbol{D}_m \dot{\boldsymbol{x}}_m \\ -\dot{\boldsymbol{x}}_m^T \boldsymbol{F}_e + \dot{\boldsymbol{x}}_m^T \boldsymbol{K}_t (\boldsymbol{x}_m - \boldsymbol{x}_r) + \frac{1}{2}\tilde{\boldsymbol{x}}_r \dot{\boldsymbol{K}}_t \tilde{\boldsymbol{x}}_r, \end{aligned} \tag{4.14}$$

where it is required to prove passivity with respect to the ports $(\dot{\boldsymbol{x}}_m, \boldsymbol{F}_h)$ and $(\dot{\boldsymbol{x}}_r, \boldsymbol{F}_e)$ which represent the ports of interaction through which the teleoperator system interacts with the operator and the remote environment. Unfortunately however, the sign of the last three terms is indefinite implying that the system can be active. In order to solve this problem, we will augment the controllers by a gobal energy tank with a state $z \in \mathbb{R}$ that exchanges energy with both the master and robot controllers, and regulates their output depending on the level of tank energy. We define our tank dynamics as

$$\begin{aligned} \dot{z} = \frac{\sigma_m \dot{\boldsymbol{x}}_m^T \boldsymbol{D}_m \dot{\boldsymbol{x}}_m}{z} + \frac{\sigma_r \dot{\boldsymbol{x}}_r^T \boldsymbol{D}_r \dot{\boldsymbol{x}}_r}{z} + \frac{\lambda_m(E(z)) \dot{\boldsymbol{x}}_m^T \boldsymbol{D}_{m,h} \dot{\boldsymbol{x}}_m}{z} \\ + \frac{\lambda_r(E(z)) \dot{\boldsymbol{x}}_r^T \boldsymbol{D}_{r,h} \dot{\boldsymbol{x}}_r}{z} - \frac{\alpha_r \dot{\boldsymbol{x}}_r^T \boldsymbol{K}_t (\boldsymbol{x}_m - \boldsymbol{x}_r)}{z} + \frac{\alpha_m \dot{\boldsymbol{x}}_m^T \boldsymbol{F}_e}{z}, \end{aligned} \tag{4.15}$$

and with tank energy $E(z) = \frac{1}{2}z^2$. As for σ_m and σ_r, they control the amount of energy filled in the tank through dissipation. They are defined as

$$\begin{cases} 0 < \sigma_r, \sigma_m \leq 1 & \text{if } E(z) < \bar{E}\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{4.16}$$

where $\bar{E}$ is an absolute upper limit for the maximum energy allowed in the tank to avoid practically unstable behavior. As for the valves α_m and α_r, they have the role of modifying the control actions if the tank is depleted. They are defined as

$$\begin{cases} 0 < \alpha_r, \alpha_m \leq 1 & \text{if } E(z) > \underline{E}\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{4.17}$$

where $\underline{E}$ is a lower threshold for the minimum energy allowed in the tank, to prevent singularities. In order to avoid complete stoppage of the task whenever the tank becomes depleted, we propose to add the valves $\lambda_m(E(z))$ and $\lambda_r(E(z))$ that have the role of harvesting additional dissipation energy to refill the tank by injecting local damping on the master and robot controllers. In order to avoid compromising task performance, this damping energy is only requested whenever the energy in the

tank drops below a threshold E_{thresh} and where $\underline{E} \leq E_{thresh} \leq \bar{E}$. $\lambda_m(E(z))$ and $\lambda_r(E(z))$ are designed as smoothly rising functions from 0 to 1, while $\boldsymbol{D}_{r,h}$ and $\boldsymbol{D}_{m,h}$ are design parameters. We now modify the master and robot controllers as

$$\begin{aligned} \boldsymbol{F}_r &= \alpha_r \boldsymbol{K}_t \left(\boldsymbol{x}_{r,d} - \boldsymbol{x}_r\right) - \boldsymbol{D}_r \dot{\boldsymbol{x}}_r - \lambda_r \boldsymbol{D}_{r,h} \dot{\boldsymbol{x}}_r, \\ \boldsymbol{F}_m &= -\alpha_m \boldsymbol{F}_e - \boldsymbol{D}_m \dot{\boldsymbol{x}}_m - \lambda_m \boldsymbol{D}_{m,h} \dot{\boldsymbol{x}}_m, \end{aligned} \tag{4.18}$$

and analyze passivity with the following storage

$$S = E(z) + \frac{1}{2} \dot{\boldsymbol{x}}_m^T \boldsymbol{M}_m \dot{\boldsymbol{x}}_m + \frac{1}{2} \dot{\boldsymbol{x}}_r^T \boldsymbol{M}_r \dot{\boldsymbol{x}}_r \,. \tag{4.19}$$

We can now make the following statement regarding the passivity of the system:

Proposition 2 *Consider the teleoperator system given by* (4.9) *and* (4.10)*, with corresponding master and robot controllers* (4.18)*. The system represents a passive map with respect to the input pair* ($\boldsymbol{F}_e$, $\boldsymbol{F}_h$) *and the output pair* ($\dot{\boldsymbol{x}}_r$, $\dot{\boldsymbol{x}}_m$)

The passivity claim can be validated by taking the rate of (4.19), which can be expressed as

$$\begin{aligned} \dot{S} = \dot{\boldsymbol{x}}_r^T \boldsymbol{F}_e + \dot{\boldsymbol{x}}_m^T \boldsymbol{F}_h - (\sigma_r - 1) \dot{\boldsymbol{x}}_r^T \boldsymbol{D}_r \dot{\boldsymbol{x}}_r - \\ (\sigma_m - 1) \dot{\boldsymbol{x}}_m^T \boldsymbol{D}_m \dot{\boldsymbol{x}}_m, \end{aligned} \tag{4.20}$$

where the last two terms are negative definite, with σ_m and σ_r set according to (4.16). Following the passivity definition in sec. 2.2.2, we can conclude that the closed-loop system is passive.

4.3.2 Simulation Study

In order to validate our passivity proof, we conducted a simulation study in Matlab® Simulink. We simulate 1 DOF bilateral teleoperation where the master and remote robots are driven by the control laws (4.18), and with an energy tank designed as explained in the previous section. We assume that the human input force is modeled by a spring-damper system that aims to follow a minimum jerk trajectory $x_{dh}(t)$, such that $F_h = k_h(x_{dh}(t) - x_h) - d_h \dot{x}_h$, where k_h and d_h are the human arm stiffness and damping, respectively, while x_h is the position of the human arm, assumed to be the same as the position of the master device x_m. We also assume that the remote robot is in contact with a viscous environment modeled by $F_e = -\mu_e \dot{x}_r$, and that the stiffness of the remote robot is modeled to be a function of the external environment force, where $K_r = 5|F_e| + 30$. Regarding the tank, we initialize it with 35 J, and we set $\bar{E} = 40$J, $\underline{E} = 0.5$J, $E_{thresh} = 8$J, $D_{r,h} = 0$ and $D_{m,h} = 20\text{N/m}^2$.
In the first scenario, we compare the performance in a setting where the human maintains a stiff grasp on the master device ($k_h = 60$N/m), thereby enhancing the stability of the system, with tank on, in the first case, and tank-off in the second. As shown in the results of Fig. 4.2(a)-4.2(c), in both cases, the system is stable and the

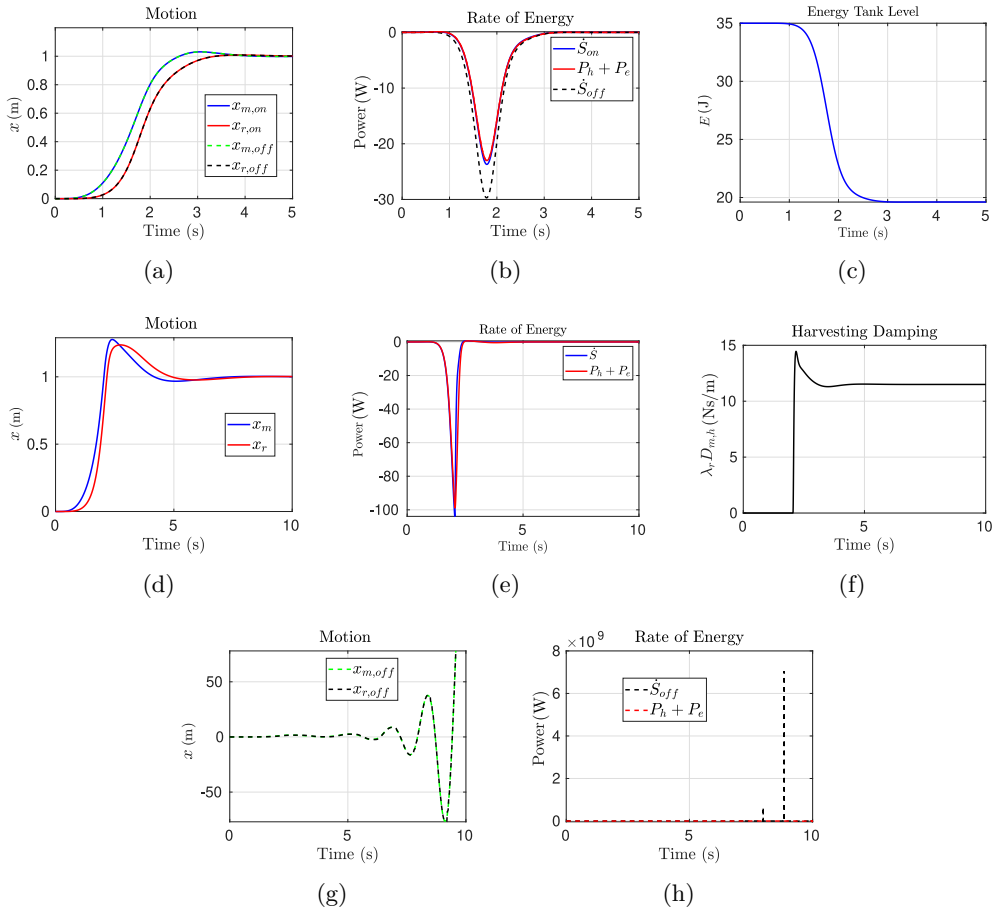


Fig. 4.2: Simulation Results for 1 DOF bilateral teleoperation System. The first row shows the results with tank on and off, when a human maintains a stiff grasp. The second row shows the results for the relaxed grasp setting with tank on, and finally, the last row shows the results with tank off, also with relaxed grasp.

performance is identical, with the remote robot following the remote robot motion. In this case, the tank does not have any effect on system performance and act only as a monitoring mechanism.

In the second scenario, we compare also the performance for tank on and tank off cases, however, in a setting where the human maintains a relaxed grasp ($k_h = 20\text{N/m}$), which is well known to induce instabilities in the system [180]. As shown in Figs. 4.2(g) and 4.2(h), the system performance without tank clearly goes unstable. On the other hand, the system remains stable with the introduction of the tank (Fig. 4.2(d)-4.2(f)).

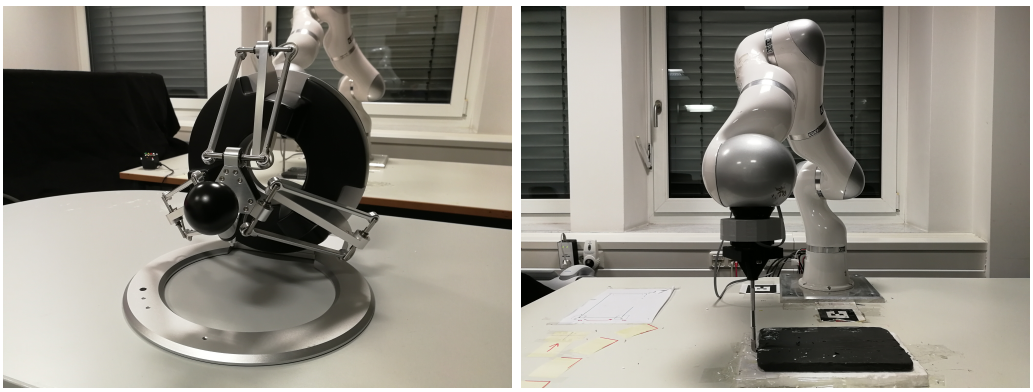


Fig. 4.3: Experimental setup used for validating our approach on a cutting task. Left: Our Omega3 haptic device used as master. Right: Our 7 DOF KUKA lightweight robot with a mounted cutting scalpel. The black clay is placed on a piece of foam on the table and serves as the material to cut. ©2021 IEEE [18]

4.4 Experimental Evaluation

To validate our approach, we conducted a series of experiments to verify both the learning and task execution. We chose cutting as a representative task. First, we briefly describe our learning phase, followed by a detailed experimental evaluation where we tested our method and compared it with constant high- and low-stiffness baselines.

As a master device, we used the Omega 3 haptic device commanded in force control mode, while the remote robot consisted of the 7 DOF KUKA lightweight robot, which was commanded in cartesian impedance control mode. For all experiments, the orientation of the robot end-effector was kept constant via active control. In order to perform the cutting task, a custom tool holder was designed, in which a cutting scalpel was fitted, and attached to the robot via a Quick-change adapter system, as shown in Fig. 4.3. A material to cut is placed on the table in front of the robot. The master device is 3 DoF and it is thus free to move along the 3 translation directions. The same constraint is imposed on the remote robot, as its orientation is fixed at the beginning of the experiment through control to make sure the scalpel is always perpendicular to the material surface. This setup is used both for the task learning and the reproduction phase. Of course, the learning part could also be done through kinesthetic teaching. However, direct interaction is not always possible with the robot in real settings (e.g., in nuclear applications). Furthermore, the physical coupling between the human and the robot induces additional dynamics during the teaching stage [125]. To avoid these drawbacks, we performed our demonstrations using direct teleoperation with haptic feedback.

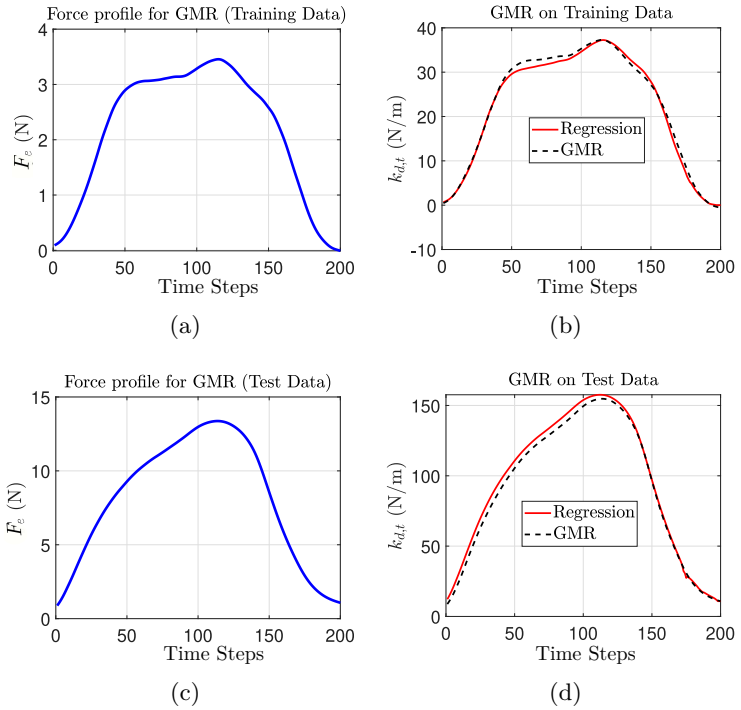


Fig. 4.4: Results of the GMR on training and test data. Left column: Force profiles used as input to the GMR. Right column: Stiffness profiles obtained via regression (red) and learnt stiffness profiles (dotted black). ©2021 IEEE [18].

4.4.1 Learning Results

For the learning phase, we conducted teleoperated cutting experiments on four materials with different stiffness properties. For each material, we collected five demonstrations. Data from all demonstrations were segmented to identify the contact phase, based on a pre-defined threshold of the contact force along the motion direction and subjected to low-pass filtering to remove sensor noise, prior to obtaining the higher order derivatives. This was followed by the stiffness estimation phase, where the algorithms described in Sec. 4.2 were run over the data. We used $k_c = 90$ N/m, while the damping was set as $b = 2\sqrt{k_c}$ N/m^2. As explained in Sec. 4.2, the k_c describes the constant low component of the human arm stiffness. Therefore, the chosen value of k_c represents a relatively low stiffness in our robotic system, and was also used for the constant low stiffness experiments. We specified a window length of $L = 3$, a regularization value $\lambda = 0.005$ and used a GMM parameterized with 4 states.

Fig. 4.4 shows the results of our learning. The right column shows the stiffness profiles obtained by GMR (dotted black) both on force data used in the training phase (upper), and on unseen force data (lower), compared to the stiffness profiles obtained via regularized regression (red). The left column shows the force profiles used

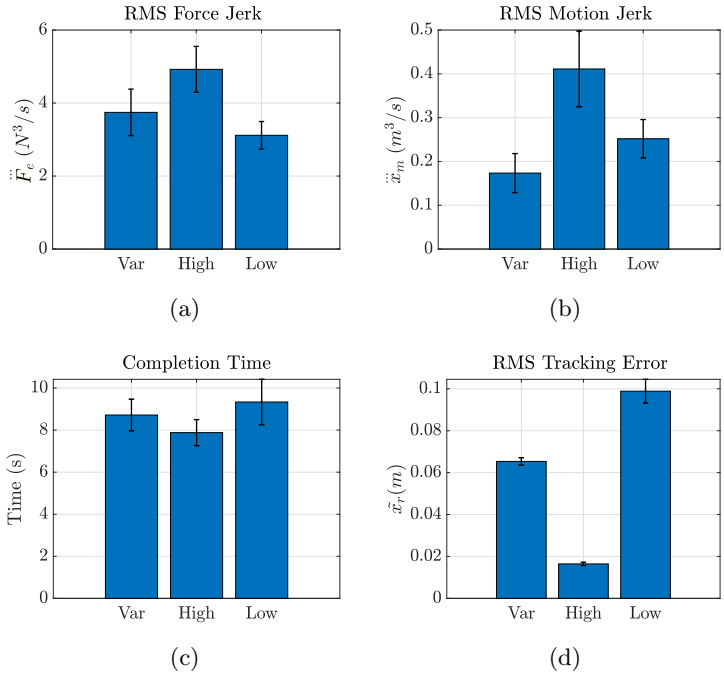


Fig. 4.5: Cutting one material, no delay. Bar plot comparison of the different metrics among variable, constant low and constant high stiffness, where the X-axis represents the condition, while the y axis represent the mean of the corresponding metric. The vertical black lines in each bar indicate the standard deviation. ©2021 IEEE [18].

as input to the regression process, which represent the mean from all demonstrations of a material.

4.4.2 Robot Validation

We validate our approach on the teleoperation system described above and shown in Fig. 4.3. The following comparisons were considered:

1) The proposed approach vs. two baseline conditions using constant low and constant high stiffness for teleoperated cutting on an unseen material with no communication delay.

2) The proposed approach vs. one baseline condition using constant high stiffness for teleoperated cutting on a given material with a communication delay of 20 ms.

3) The proposed approach vs. one baseline condition using constant low stiffness for teleoperated cutting through two different materials with no communication delay.

Cutting one material, no delay

In the first series of experiments, we aim to test our proposed solution when cutting a material which was not used during the demonstrations. We compare our approach against baseline executions of constant low stiffness (90 N/m) and constant high stiffness (800 N/m). For each condition, fifteen trials were conducted where a human operator commands the remote robot with similar motions. For fairness of comparison, the depth of cut was not controlled by the user but was maintained constant in all trials. We compare the three conditions with respect to the following performance metrics:

- Root Mean Square (RMS) of motion tracking error between the master interface and the remote robot;
- Task completion time;
- RMS of the master robot motion jerk;
- RMS of the remote robot force jerk.

While the first two metrics are an indication of the task execution speed, we choose the last two metrics as a measure of smoothness in the task execution. Jerk metrics have also been used to characterize physical fatigue levels [241]. The results of this study are shown in Fig. 4.5 as a bar plot showing the mean of the analyzed metrics over trials, together with the standard deviation. As expected, the RMS tracking error is the highest for the low stiffness case, with a mean across trials of 0.0989 ± 0.0057 m compared to a mean of 0.0164 ± 0.0008 m for the high stiffness case and 0.0654 ± 0.0018 m for the proposed variable case. Similarly, the completion time is highest in the low stiffness case (9.3359 ± 1.0792 s) when compared to the variable stiffness and high stiffness cases, which have relatively lower completion times of 8.7171 ± 0.7522 s and 7.8807 ± 0.6193 s. On the other hand, in terms of smoothness, our variable stiffness approach outperforms the high stiffness case. The former shows a RMS motion jerk and RMS force jerk of 0.1735 ± 0.0446 $ms^{-3}$and 3.7448 ± 0.6368 Ns^{-3}, respectively, compared to 0.4112 ± 0.0863 ms^{-3} and 4.9233 ± 0.6291 Ns^{-3} for the high stiffness case.

Cutting one material, with delay

The difference in performance between the variable and high stiffness cases can be better seen in the presence of disturbances, e.g., communication delays. It is well known that for force reflecting teleoperation, induced master motion can result in violent recoiling of the remote robot during contact, a problem further aggravated with delays. Therefore, we conducted a representative experiment where an artificial delay of 20 ms between master and remote systems was simulated. This delay was chosen based on typically used values in the literature on delayed bilateral teleoperation [239]. In this case, we kept the energy tank off, allowing us to showcase the difference in performance/safety solely from the considered stiffness method.

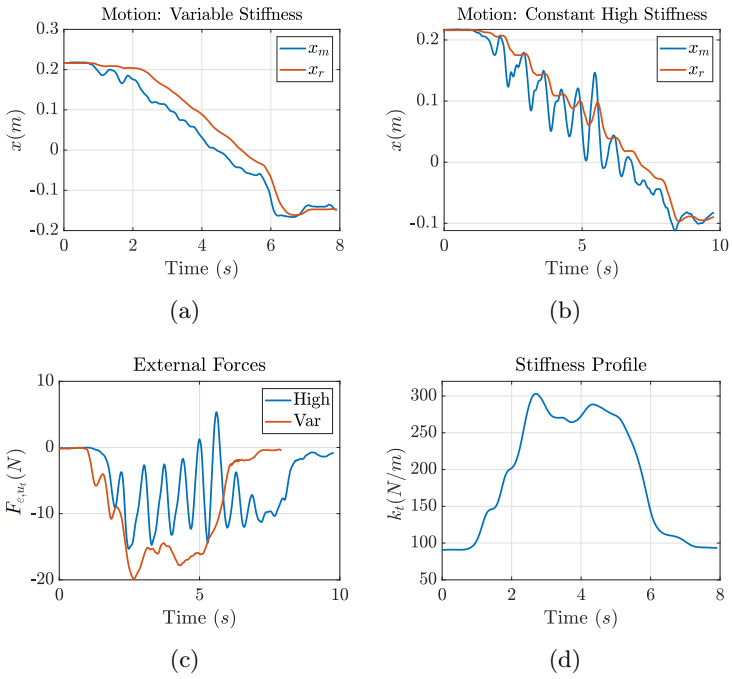


Fig. 4.6: Cutting one material, with delay. The upper row shows the motion trajectories of the master (blue) and remote manipulator (orange) both for the variable case (left) and the high stiffness case (right). The left-bottom figure shows the sensed external forces for the high stiffness case (blue) and for the variable case (orange), while the bottom right figure shows the stiffness profile for the variable stiffness. All variables are displayed along the axis of motion. ©2021 IEEE [18].

Results are shown in Fig. 4.6. It depicts the motion profiles of the master and the remote robot, for both the variable and the constant high stiffness cases (upper row), the external forces recorded using the robot force sensors, and the stiffness profile for the variable stiffness case. Clearly, for the high stiffness case, the task becomes quickly unstable, inducing high oscillations in the motion and force profiles, endangering both the task performance and the robot safety. As expected, for the variable stiffness case, oscillations are significantly smaller in the motion, force and stiffness profiles, indicating a much smoother task execution. It is also worth noting that execution time is also shorter for the variable case, where the task is completed in around 1.8 s less.

Cutting two materials, no delay

While the above situation shows the merit of our method compared to constant high stiffness in terms of safety, here we analyze the benefit of our approach compared to a constant low stiffness approach. More specifically, we test our approach in a situation where it is required to cut through a piece composed of a soft part and then a hard part. The results of this experiment are shown in Fig. 4.7. On the right, it shows the stiffness profile for the variable stiffness case. On the left, it shows the

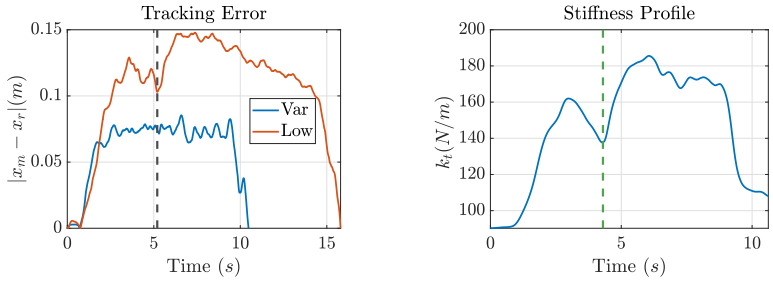


Fig. 4.7: Cutting two materials, no delay. Results of the transition from soft to hard materials while cutting. The left figure shows the tracking error for the low (orange) and variable (blue) stiffness case. The right figure shows the variable stiffness profile generated from learnt GMM. The dotted vertical lines denote the moment of transition from soft to hard, where the black line indicates the moment of transition for the low stiffness case ($t \approx 5.2$), while the dotted green denotes the transition for the variable case ($t \approx 4.2$). For better readability of the graphs, in the left figure we show the moment of transition for the low stiffness case only. ©2021 IEEE [18].

absolute tracking error between the master and robot motion, along the y axis (the motion direction), both for the variable stiffness case (blue) and the low stiffness case (orange). The stiffness profile starts increasing smoothly at the moment of transition, at $t \approx 4.2$ s, to account for the increased hardness of the materials, which is reflected in higher sensed external forces. This adaptation results in the motion tracking error between master and remote robot to remain almost constant throughout the task. On the other hand, in the low stiffness case, the tracking error becomes higher after the transition at $t \approx 5.2$ s, due to the inability of the robot controller to adapt to the higher resistance force of the harder material.

4.5 Discussion

Previous works have analyzed the potential benefits of incorporating variable impedance control into a teleoperation architecture. Our experimental results further consolidate these findings. In fact, while using a constant high stiffness (with respect to the environment) results in better tracking accuracy and faster execution, it poses potentially-dangerous stability issues. On the other hand, using a constant low stiffness results in degraded tracking accuracy and higher completion times, but guarantees higher safety margins. Our approach aims at combining the best of the two worlds. By giving preference to a low stiffness that increases only when needed, we are able to achieve a good compromise between task execution speed, tracking accuracy, smoothness, and safety. Compared to previous works on the topic, our system does not need any cognitive effort from the user, as the stiffness is chosen automatically at runtime, or any additional sensor on the user or robot sides.

The efficiency of our method was confirmed in the first experimental study, where

we achieved task completion times comparable to those of the high stiffness method but with the advantage of enhanced smoothness, both on motion and force levels. This difference can be attributed to the oscillations resulting from the induced master motion, which become higher with a high stiffness gain, making it harder for the human operator to maintain a stable and smooth contact. This problem becomes significantly aggravated in presence of a delay in the communication channel. As expected, our approach is much more robust in this regard, even without applying any stability related-solutions dedicated for delayed bilateral telemanipulation. It could be argued that a slightly lower constant high stiffness might improve the performance. However, the choice of this parameter to provide a compromise between accuracy and robustness is task specific and not always easy to determine. On the contrary, our algorithm determines this desired stiffness automatically. Furthermore, we validated the ability of our approach to adapt to situations that feature force variations typical in a transition from a soft to hard material in cutting, or variations induced by a gravitational load during a lifting task [7]. The stiffness increases smoothly when the hard material is detected, maintaining the tracking error constant and not requiring the operator to exert any additional effort to compensate for it by, e.g., stiffening up his/her arm. This result implies that the human operator can focus on the careful execution of the task, without any distraction, while the robot automatically adapts its interaction force with the environment. It is worth noting that such an adaptation is made possible thanks to the fact that our GMM is not time-driven, but instead it performs the regression on the perceived external forces, alleviating the necessity of any additional vision module.

4.6 Summary

In this chapter, we presented a variable impedance teleoperation system for contact tasks. It automatically adapts to the environment by changing the stiffness of the remote robot. We use multiple demonstrations on different materials to learn the stiffness model with respect to the applied force at the remote environment. This variable stiffness estimation at run-time is then incorporated in the teleoperation architecture and allows the robot to perform the task with enhanced robustness, and further adapt to new environments. We performed a passivity analysis on our system and ensure its passivity using energy tanks. We then tested our algorithm in three cutting tasks, including a situation with communication delay and one where the robot had to cut through two different materials. Results showed the advantages of our approach compared to a constant high or a constant low stiffness.

CHAPTER 5

Shared Control For Contact Tasks

This Chapter presents a novel shared control architecture dedicated to teleoperated contact tasks. We use Learning from Demonstration as a framework to learn a task model that encodes the desired motions, forces and stiffness profiles. Then, the learnt information is used by a Virtual Fixture (VF) to guide the human operator along a nominal task trajectory that captures the task dynamics, while simultaneously adapting the remote robot impedance. Furthermore, we provide haptic guidance in a human-aware manner. To that end, we propose a control law that eliminates time dependency and depends only on the current human state, inspired by the path and flow control formulations used in the exoskeleton literature [207, 242]. The proposed approach is validated in a user study where we test the guidance effect for the bilateral teleoperation of a drawing and a wiping task. The experimental results reveal a statistically significant improvement in several metrics, compared to teleoperation without guidance. The contents of this chapter are based on the following publication [19]

Y. Michel, Z. Lee and D. Lee, "A Learning-Based Shared Control Approach for Contact Tasks," in IEEE Robotics and Automation Letters, vol. 8, no. 12, pp. 8002-8009, 2023.

5.1 Motivation

In this chapter, we propose a shared control approach targeted for contact tasks. Imagine a scenario where we need to perform a teleoperated drawing or carving task following a certain shape. A biomimetic strategy of performing these tasks would be to follow the motion trajectory according to the desired path. Additionally, we would also adapt the endpoint impedance to counteract disturbances caused by friction or surface irregularities, while also maintaining a suitable downward force in

order to apply pressure on the surface [3, 8]. Therefore, we argue that in order to provide guidance for a human performing a teleoperated task that features continuous physical contact with the environment, the Autonomous Agent (AA) should encode all relevant information for the robot to appropriately shape its behavior during contact. In addition to the nominal motion, this would also include desired forces, task frames, and impedance profiles.
In the previous chapter, we took a first step in this direction where we proposed an AA that adapts the remote robot endpoint stiffness during teleoperated contact tasks, where the adaptation is dictated by the state of the real-time interaction with the remote environment. In this chapter, we extend our work by additionally encoding desired motions, task frames and forces in the AA, therefore shaping all the relevant features for a robot performing contact.
Similar to the works reviewed in section 2.1.5, we also consider tasks that involve the remote manipulator being physically coupled to its environment while performing the task. However, the form of guidance that was considered in these works focused mainly on the motion aspect of the task, without any consideration to other potentially important aspects for a robot in contact, such as the desired forces the remote robot should apply and the required impedance. In addition to that, the developed techniques in the aforementioned literature are tailored to suit the considered task requirements, and therefore it is difficult to generalize these approaches to other applications. On the other hand, in this work, we propose a general pipeline by exploiting LfD to learn a task model from user-provided demonstrations.
In fact, as reviewed in section 2.1.5, several shared control works considered also LfD for guidance generation. A fundamental difference in our case is that in addition to the motion profile, we encode information within our learning model related to task directions that require the application of forces, and a stiffness adaptation policy. Thereby, we endow our AA with means to assist the human operator in all the relevant aspects for a robot performing physical interactions. A block diagram illustrating our approach is shown in Fig. 5.1.

5.2 Task Learning

In this section, we present our learning-based shared control architecture, where a human collaborates with an AA in order to perform a task. Since we mainly target contact tasks, it becomes reasonable that the AA encodes knowledge regarding desired motions, forces, task frames and a variable stiffness profile. In the following, we will explain our learning procedure on how to retrieve such information from the underlying task dynamics captured in user demonstrations. Then, the extracted information is encoded in a probabilistic model that can subsequently be used for regression. This model is then exploited by the AA to provide haptic feedback, generating guidance forces that can assist the user to achieve the desired task. In the following, we use the notation $\boldsymbol{\lambda}_i(\boldsymbol{A})$ to indicate the i-th eigen value of an arbitrary matrix $\boldsymbol{A}$.

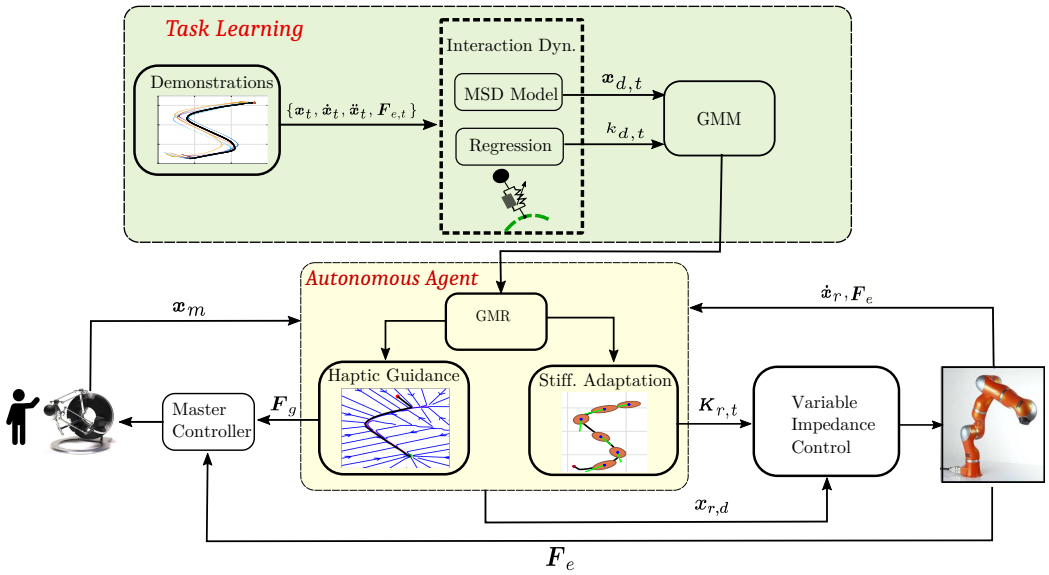


Fig. 5.1: A block diagram illustrating the overall architecture of our system. The Block "Task Learning" highlighted in light green depicts the procedure to learn a model that fully captures the task dynamics via a Mass-Spring-Damper (MSD) model for subsequent usage during task execution. The Autonomous Agent (light yellow) then uses the encoded model during teleoperation in order to provide haptic guidance to the human operator in real-time, while also adapting the remote robot stiffness based on the measured robot states.

5.2.1 Task Dynamics

For learning, an expert user provides a set of demonstrations where we collect the robot end effector positions $\{\boldsymbol{x}_t\}_{t=1}^{T}$ and the sensed external forces at the end effector $\{\boldsymbol{F}_{e,t}\}_{t=1}^{T}$, with t as the time instant and T the length of each demonstration. We can then obtain the velocities $\{\dot{\boldsymbol{x}}_t\}_{t=1}^{T}$ and accelerations $\{\ddot{\boldsymbol{x}}_t\}_{t=1}^{T}$ via numerical differentiation.

As stated earlier, we aim to learn a regression model that encodes motion, stiffness and force profiles that represents the AA guiding the human operator. Similar to the previous chapter, we approximate the observed dynamics during demonstrations with a unit-mass driven by a spring and damper, subject to external forces from the environment. For the general multi-DOF case, this can be expressed as $\ddot{\boldsymbol{x}}_t = \boldsymbol{K}_t(\boldsymbol{x}_{t,d} - \boldsymbol{x}_t) - \boldsymbol{D}_t\dot{\boldsymbol{x}}_t + \boldsymbol{F}_{e,t}$ with $\boldsymbol{K}_t$ and $\boldsymbol{D}_t$ as the positive definite, possibly varying stiffness and damping matrices, and $\boldsymbol{x}_{t,d}$ is the unknown attractor path of the spring. This can be further simplified as

$$\ddot{\boldsymbol{x}}_t = \boldsymbol{K}_c(\boldsymbol{x}_{t,d} - \boldsymbol{x}_t) - \boldsymbol{D}\dot{\boldsymbol{x}}_t. \tag{5.1}$$

using the decomposition $\boldsymbol{K}_t = \boldsymbol{K}_c + \boldsymbol{K}_{d,t}$ following the reasoning explained in section 4.2.

Previously, we computed $\boldsymbol{x}_{t,d}$ by fixing $\boldsymbol{K}_c$ and inverting the MSD dynamics with the constant stiffness, which can then be used for obtaining $\boldsymbol{K}_{d,t}$ via regression. This solution however works only for task directions that feature both motion and contact

e.g during cutting. For purely force controlled directions, where $\ddot{\boldsymbol{x}}_t \approx \dot{\boldsymbol{x}}_t \approx 0$[1], this reasoning would not work since we would get $\boldsymbol{x}_t \approx \boldsymbol{x}_{t,d}$ when computing $\boldsymbol{x}_{t,d}$, and in consequence the subsequent regression would fail.
To solve this problem, we propose a solution inspired by the hybrid motion-force control literature [37]. More specifically, we can derive for each time instant an orthogonal matrix $\boldsymbol{R}_t \in \mathbb{R}^{3\times3}$ that belongs to $SO(3)$ representing the rotation of the task frame (i.e constraint frame) with respect to the base frame of the robot at the current time instant. We can also specify $\boldsymbol{S} \in \mathbb{R}^{3\times3}$ as a diagonal selection matrix with 0 entries for the motion controlled directions, and 1 for the force controlled direction. In our case, we assign the force control dimension arbitrarily to the $z-$ axis of the task frame. This corresponds to setting the third eigen value of $\boldsymbol{S}$ to 1 i.e $\boldsymbol{\lambda}_3(\boldsymbol{S}) = 1$. We can then compute a projection matrix $\boldsymbol{U}_t = \boldsymbol{R}_t\boldsymbol{S}\boldsymbol{R}_t^T$ that rotates the selection matrix with the current task frame. In order to compute $\boldsymbol{R}_t$, we set the first eigen vector to be along the unit vector of the current velocity direction i.e $\vec{\boldsymbol{v}}_t = \frac{\dot{\boldsymbol{x}}_t}{||\dot{\boldsymbol{x}}_t||}$, while the second and third directions are derived via Gram-Schmidt orothognalization to form a right-handed frame.
We can then proceed to compute the attractor path by inverting (5.1) while taking into consideration $\boldsymbol{U}_t$ via

$$\boldsymbol{x}_{t,d} = (\boldsymbol{I} - \boldsymbol{U}_t)\boldsymbol{K}_c^{-1}(\ddot{\boldsymbol{x}}_t + \boldsymbol{D}\dot{\boldsymbol{x}}_t) + \boldsymbol{U}_t\boldsymbol{K}_c^{-1}\boldsymbol{F}_{e,t} + \boldsymbol{x}_t\,. \tag{5.2}$$

For the simple case where $\boldsymbol{R}_t$ coincides with the world frame, (5.2) yields $\boldsymbol{x}_{t,d} = \boldsymbol{K}_c^{-1}(\ddot{\boldsymbol{x}}_t + \boldsymbol{D}\dot{\boldsymbol{x}}_t) + \boldsymbol{x}_t$ for motion controlled directions and $\boldsymbol{x}_{t,d} = \boldsymbol{K}_c^{-1}\boldsymbol{F}_{e,t} + \boldsymbol{x}_t$ for the force controlled direction. Intuitively, for the motion controlled subspace, the attractor trajectory follows closely the observed end effector motion, while in the orthogonal force subspace, the attractor represents a penetration into the surface with a profile dictated by the external force along the $z-$ axis of the task frame.

Stiffness Extraction

As stated earlier, we assume that stiffness adaptation only occurs along the direction of motion, where the policy is to keep the stiffness low, and adapt it only needed to compensate for the environmental disturbance, reflected by the sensed external forces in this direction. It is important to note here that depending on the task direction, the external forces convey different types of information. In motion-controlled directions, the external forces are related to disturbances such as friction that need to be compensated, while in the force-controlled subspace, external forces represent a control goal that should be reproduced by the robot.
We define $\tilde{\boldsymbol{X}}_t = \boldsymbol{x}_{t,d} - \boldsymbol{x}_t$ and $\tilde{\boldsymbol{Y}}_t = \boldsymbol{K}_c^{-1}(\ddot{\boldsymbol{x}}_t + \boldsymbol{D}\dot{\boldsymbol{x}}_t + \boldsymbol{F}_{e,t})$. We generalize the formulation from Chapter 4 by projecting $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$ along the motion direction obtaining $\tilde{x}_{t,\vec{v}_t} = \tilde{\boldsymbol{X}}_t.\vec{\boldsymbol{v}}_t$ and $\tilde{y}_{t,\vec{v}_t} = \tilde{\boldsymbol{Y}}_t.\vec{\boldsymbol{v}}_t$, with $\vec{\boldsymbol{v}}_t = \frac{\dot{\boldsymbol{x}}_t}{||\dot{\boldsymbol{x}}_t||}$ being the unit vector of the velocity. We can then compute the varying component $k_{d,t}$ via regularized regression over a sliding time window as

$$k_{d,t} = (\tilde{\boldsymbol{X}}_{t,\vec{v}_t}^T\tilde{\boldsymbol{X}}_{t,\vec{v}_t} + \epsilon\boldsymbol{I})^\dagger(\tilde{\boldsymbol{X}}_{t,\vec{v}_t}\tilde{\boldsymbol{Y}}_{t,\vec{v}_t}^T), \tag{5.3}$$

[1]The implicit assumption that the environment is perfectly rigid is made

where $\tilde{\boldsymbol{X}}_{t,\vec{v}_t} = [\tilde{x}_{t-L,\vec{v}_t} \dots \tilde{x}_{t+L,\vec{v}_t}]$ and $\tilde{\boldsymbol{Y}}_{t,\vec{v}_t} = [\tilde{y}_{t-L,\vec{v}_t} \dots \tilde{y}_{t+L,\vec{v}_t}]$ represent a concatenation of values over the window L, while $\epsilon > 0$ is a small constant for regularization.

5.2.2 Task Encoding

With the above procedure, we augment our data with a new data set $\{\boldsymbol{x}_{t,d}, k_{d,t}\}_{t=1}^{T}$ that fully captures the task dynamics in terms of observed motions, forces and stiffness profiles. We still however need to encode this data into a compact representation that describes an input-output relationship which can be utilized to perform regression. To that end, we also use GMM (please refer to sec. 2.2.5) to learn the joint distribution of $\boldsymbol{\xi} = [\boldsymbol{\xi}_i, \boldsymbol{\xi}_o]^T$. For our problem here, we set $\boldsymbol{\xi} = [t, \boldsymbol{x}_{t,d}]^T$ for encoding the attractor path which outputs a desired position for each time instant. On the other hand, for stiffness, we use $\boldsymbol{\xi}_i = [F_{e,\vec{v}_t}, ||\dot{\boldsymbol{x}}_t||]$ as input data and $\xi_0 = k_{d,t}$ as output, where $F_{e,\vec{v}_t}$ indicates the projected external force along the velocity direction, which mainly reflects disturbances such as friction. Note that, while in Chapter 4 we use $\xi_i = F_{e,\vec{v}_t}$, we include here the norm of the velocity $||\dot{\boldsymbol{x}}_t||$ as an additional input feature, inspired by the recent results of [103] for the execution of dynamic contact tasks.

5.3 Shared Control Architecture

In the following, we present our shared control approach, where we consider a bilateral teleoperation scenario. In particular, we use the learnt task model in the previous section in order to shape the impedance behavior of the remote robot, while simultaneously generating haptic assistance on the master robot to guide the human to the learnt attractor path.

5.3.1 Remote Robot Control

The goal of the control for the remote manipulator is to follow a desired motion command. This is achieved with a variable impedance control law of the following form

$$\boldsymbol{F}_r = \boldsymbol{K}_{r,t} \left(\boldsymbol{x}_{r,d} - \boldsymbol{x}_r\right) - \boldsymbol{D}_{r,t}\dot{\boldsymbol{x}}_r, \tag{5.4}$$

where $\boldsymbol{x}_{r,d}$ is the desired reference for the remote robot, and will be further specified in the following. The state-varying stiffness $\boldsymbol{K}_{r,t}$ is constructed in a manner similar to section 4.2.3, with $\boldsymbol{\lambda}_1(\boldsymbol{K}_{r,t}) = \boldsymbol{\lambda}_1(\boldsymbol{K}_c) + \hat{k}_{d,t}$ with $\hat{k}_{d,t}$ computed via GMR based on the current remote robot state $(\dot{\boldsymbol{x}}_r, \boldsymbol{F}_e)$, while $\boldsymbol{\lambda}_j(\boldsymbol{K}_{r,t}) = \boldsymbol{\lambda}_j(\boldsymbol{K}_c)$ for $j = \{2, 3\}$, and with eigen vectors pointing along and perpendicular to the current motion direction.

5.3.2 Haptic guidance

The goal of control on the other hand on the master interface is two-fold: first, to provide force feedback based on the external forces $\boldsymbol{F}_e$ at the remote robot end-effector, allowing the human operator to sense the environment. Second, we aim to assist the human operator by providing her/him haptic guidance along a desired

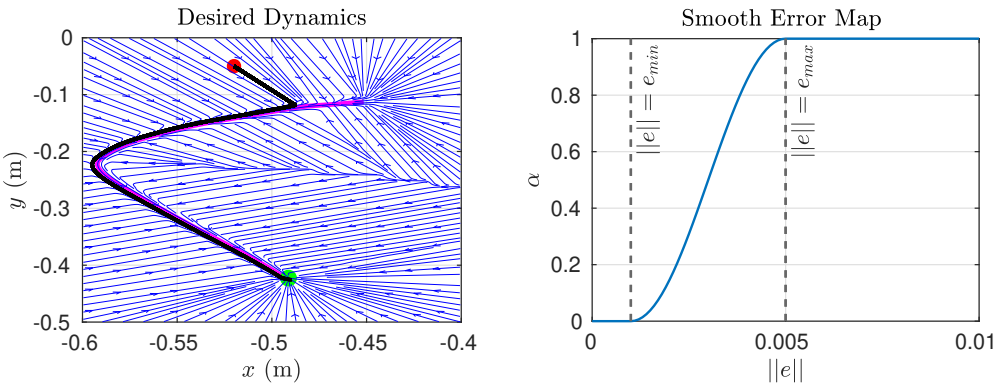


Fig. 5.2: Left: The figure shows the streamlines of the dynamics governing the haptic guidance generation in blue, along with the desired path for haptic guidance shown in pink. The actual simulated robot motion is shown in black, with the red dot showing the initial robot position, while the green dot is final position. Right: A figure depicting the smooth error map function α of equation (5.6), where e_{min} and e_{max} are user defined thresholds that dictate the start and end of the transitions from the flow to the path controller. ©2023 IEEE [19]

motion.
To that end, we propose the following control law for the master interface

$$\boldsymbol{F}_m = -\boldsymbol{W}_e\boldsymbol{F}_e - \boldsymbol{D}_m\dot{\boldsymbol{x}}_m + \boldsymbol{W}_g\boldsymbol{F}_g, \tag{5.5}$$

where $\boldsymbol{W}_e$ and $\boldsymbol{W}_g$ are additional weights that will be further specified in the following. The guidance forces are denoted $\boldsymbol{F}_g$, that will be designed based on the attractor path retrieved via GMR, denoted $\hat{\boldsymbol{x}}_{t,d}$. Note however that $\hat{\boldsymbol{x}}_{t,d}$ is represented in the remote robot end-effector frame, where the goal of the task is typically expressed, and should be first transformed to the master-interface. This is achieved via $\hat{\boldsymbol{x}}^m_{t,d} = \boldsymbol{H}\boldsymbol{\beta}\hat{\boldsymbol{x}}_{t,d}$, where $\hat{\boldsymbol{x}}^m_{t,d}$ is the desired attractor path expressed in the end-effector of the master interface, $\boldsymbol{H} \in \mathbb{R}^{4\times4}$ is a homogeneous transformation between the two coordinate frames, while $\boldsymbol{\beta}$ is a scaling factor that accounts for possible workspace differences. One possible way to design $\boldsymbol{F}_g$ is to simply use an impedance control law that follows $\hat{\boldsymbol{x}}^m_{t,d}$ such that $\boldsymbol{F}_g = \boldsymbol{K}_m(\hat{\boldsymbol{x}}^m_{t,d} - \boldsymbol{x}_m) - \boldsymbol{D}_m\dot{\boldsymbol{x}}_m$ with stiffness $\boldsymbol{K}_m$ and damping $\boldsymbol{D}_m$. This solution, however, is time-based which means that it completely ignores the current operator state, and therefore does not allow any timing freedom in the task execution. To remedy this problem, a more elaborate solution is proposed in the following.

Guidance Control Law

In the exoskeleton literature, haptic guidance is a viable tool to provide assistance for human subjects during walking. In this regard, two prominent control strategies can be distinguished: path [242] and flow Control [207]. In the former, the main idea is to derive the guidance from the gradient of a potential field, effectively implementing a spring and damper that provide restoring forces to the desired path. On the other

hand, the flow controller provides guidance derived from a flow field, by regulating a velocity error through a positive damping gain. Inspired by these approaches, we present a new formulation that suits our desiderata. In regions far away from the desired path, the guidance should pull the human back to the desired path, similar to the path control law. On the other hand, when the user is close enough to the path, we aim to provide assistance to pull the user forward along the velocity streamlines of the desired path i.e. flow control law. It is important also that the transition between the two regimes is smooth.
Let $\boldsymbol{x}_{c,md}$ be a position on the path closest to the current position of the master interface, and $\dot{\boldsymbol{x}}^m_{c,d}$ be the corresponding desired velocity, we propose the following control law

$$\boldsymbol{F}_g = \begin{cases} \boldsymbol{F}_p = \boldsymbol{K}_g(\boldsymbol{x}^m_{c,d} - \boldsymbol{x}_m) - \boldsymbol{D}_g\dot{\boldsymbol{x}}_m & \text{if } ||\boldsymbol{e}|| > e_{max} \\ \boldsymbol{F}_f = \boldsymbol{D}_f(\dot{\boldsymbol{x}}^m_{c,d} - \dot{\boldsymbol{x}}_m) & \text{if } ||\boldsymbol{e}|| < e_{min}\,, \\ \alpha(||\boldsymbol{e}||)\boldsymbol{F}_p + (1-\alpha(||\boldsymbol{e}||))\boldsymbol{F}_f & \text{otherwise}\ , \end{cases} \tag{5.6}$$

where $\boldsymbol{e} = \boldsymbol{x}^m_{c,d} - \boldsymbol{x}_m$, while $\boldsymbol{e}_{max}$, $\boldsymbol{e}_{min}$ are user-defined thresholds for the maximum and minimum errors allowed with respect to the path. The smooth transition between the path and flow control modes is provided by the continuous function $\alpha(||\boldsymbol{e}||)$, depicted in Fig. 5.2, right. $\boldsymbol{K}_g$ and $\boldsymbol{D}_g$ denote the constant positive definite stiffness and damping matrices for the path control law, while $\boldsymbol{D}_f$ is the damping gain of the flow controller.
The choice of the thresholds $\boldsymbol{e}_{max}$ and $\boldsymbol{e}_{min}$ depend on the degree of precision required for the task: high values for e_{min} and e_{max} mean that the flow controller is activated for larger values of $||\boldsymbol{e}||$, and therefore provides assistance along the flow field in a wider area around the reference path. This results in following the overall shape of the movement, but with larger deviations from the nominal path.
While [207] used a constant gain for the flow controller, we opted here for a design inspired by [131], where the Damping matrix provides a force along the desired motion direction, while selectively dissipating energy in directions perpendicular to it. This gives the eigen values of $\boldsymbol{D}_f$ a nice physical interpretation, where the first eigen value tunes the strength with which the user is pulled along the path, while the remaining eigen values penalize deviations from the nominal motion. Fig. 5.2 shows example streamlines of the dynamics imposed by the controller and the position response of a unit mass subject to the controller action. Since the starting position (red dot) is far from the nominal path shown in pink, the mass is initially driven by the path control law. Once the mass is close enough to the path, it is driven by the flow controller guiding it to the final goal position, shown as a green dot.

5.3.3 Guidance Modes

In the following, we outline more specifically the master and remote robot controllers used during operation. To that end, we propose two shared control architectures.

Guidance Mode 1

In the first guidance mode, we chose a straightforward conventional haptic shared control architecture [203], where the guidance forces (i.e. virtual fixtures) are implemented on the master robot to guide the human along the nominal task trajectory, while the human is fully in charge of controlling all DOF of the remote robot. Furthermore, the guidance forces and the reflected forces from the remote environment are superimposed in all the DOF. To summarize, we set $\boldsymbol{x}_{r,d} = \boldsymbol{x}_m$, while the weights $\boldsymbol{W}_e$ and $\boldsymbol{W}_g$ are set to the identity matrix i.e. $\boldsymbol{W}_e = \boldsymbol{W}_g = \boldsymbol{I}_3$.
It can be easily noticed here that for the motion controlled directions, a human operating the master robot is guided to the path described by $\hat{\boldsymbol{x}}^m_{t,d}$ that would lead to a remote robot motion that resembles the demonstrated ones. On the other hand, for the force controlled direction, the haptic cues guide the user to a path that follows the demonstrated force profile (cf. (5.2)). On the remote robot, this translates into a surface penetration which together with the stiffness setting in section 5.3.1, the robot applies indirectly the desired force effectively via the spring action of the impedance control law (5.4).

Guidance Mode 2

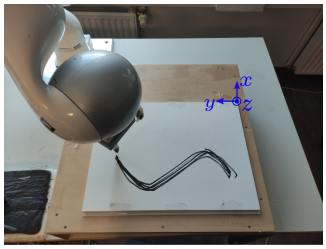
The potential drawback in the Guidance mode 1 is the possible conflict between the guidance force and the force feedback, which could diminish the user ability to properly feel the guidance. In the second mode, we choose a smarter structure for the shared control that exploits information about the task frames. More specifically, we opt for a partitioned space architecture where the human is in charge of the motion controlled directions, while the autonomy takes care of the force controlled direction. We follow a similar intuition for the master robot control, where the haptic guidance is provided in the motion controlled direction, while the force feedback from the environment is set in the force-controlled direction. To implement this reasoning, we use

$$\boldsymbol{x}_{r,d} = (\boldsymbol{I} - \boldsymbol{U}_t)\boldsymbol{x}_m + \boldsymbol{U}_t\boldsymbol{x}^r_{c,d} \tag{5.7}$$

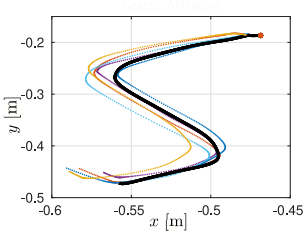
$$\boldsymbol{W}_g = \boldsymbol{I} - \boldsymbol{U}_t \tag{5.8}$$

$$\boldsymbol{W}_e = \boldsymbol{U}_t \tag{5.9}$$

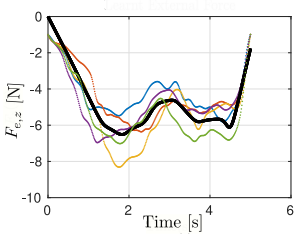
where $\boldsymbol{x}^r_{c,d} = \boldsymbol{H}^{-1}\frac{1}{\beta}\boldsymbol{x}^m_{c,d}$ is the mapping of $\boldsymbol{x}^m_{c,d}$ in the remote robot frame, while the matrix $\boldsymbol{U}_t$ is computed as in section 5.2.1, and with $\boldsymbol{R}_t$ evaluated from the current motion direction of the remote robot. The role of $\boldsymbol{U}_t$ in the above equations is such that in the motion controlled direction, the desired remote robot motion is provided by the operator i.e. $\boldsymbol{x}_m$, while the AA autonomously controls the robot in the force controlled direction, aiming to regulate the external force in the $z-$ direction according the desired one. This becomes possible since the attractor path in the $z-$ direction follows the force profile the remote robot should apply to the environment, as explained earlier. On the master side, the role of $\boldsymbol{U}_t$ is to provide feedback relevant to the task direction i.e. a guidance force in the motion-direction, and the environment force in the orthogonal force-controlled subspace.



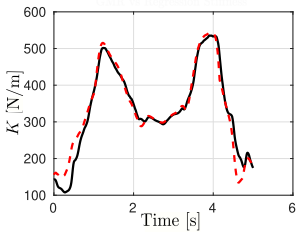
(a) Robot Drawing Setup



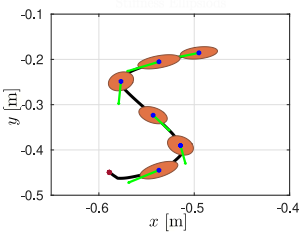
(b) Learnt Attractor



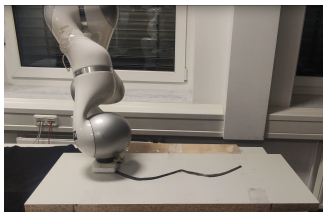
(c) Learnt Force Profile



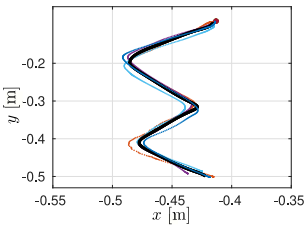
(d) Learnt Stiffness



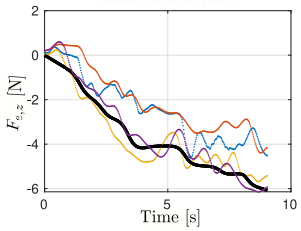
(e) Stiffness Ellipsoids



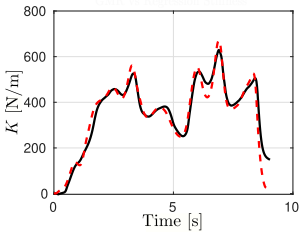
(f) Robot Wiping Setup



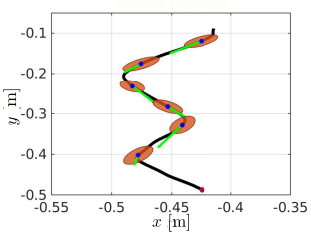
(g) Learnt Attractor



(h) Wiping



(i) Learnt Stiffness



(j) Stiffness Ellipsoids

Fig. 5.3: Learning Results of the considered drawing and wiping tasks. Fig. 5.3(a)-5.3(e) show the results for the drawing, while Fig. 5.3(f)-5.3(j) are the wiping results. Fig. 5.3(a) and 5.3(f) show the robot setup. Fig. 5.3(b),5.3(g) illustrate the demonstrated spatial trajectories in the $x-y$ plane, and in thick black the retrieved attractor path via GMR. Similarly, Fig. 5.3(c),5.3(h) highlight the demonstrated force profiles and the reconstructed force based on the attractor path retrieved via GMR, in the $z-$plane. Fig. 5.3(d),5.3(i) show the stiffness profiles obtained by regression (dotted red) compared to the profiles retrieved via GMR (solid black). Finally, the Fig. 5.3(e),5.3(j) depict the reconstructed stiffness ellipsoids along the attractor path. The green arrow shows the current velocity direction. ©2023 IEEE [19]

5.4 Experimental Evaluation

In this section, we present the experimental validation of our proposed framework. First, we describe our procedure for collecting demonstrations and learning task models, then we validate our shared control approach in a teleoperated drawing task of an S-shape, and a wiping task following an M-Shape. For drawing, The robot was mounted with a chalk marker for drawing on its end-effector. Note that for such task, besides reproducing the motion to draw the desired shape, the force profile applied by the robot is also crucial: The drawing marker works by applying pressure on a paper, and therefore a certain force is necessary, however, applying too much force can eventually lead to breaking the marker, or result in a high friction that makes it very hard to move. Furthermore, it is necessary to always maintain a minimum level of force to ensure the marker remains in contact with the paper. A similar requirement applies for the wiping, where a minimum force is necessary to ensure the correct wiping of the surface. The need to concurrently control both motions and forces can render teleoperation difficult for such tasks, and therefore, shared control would be useful for assisting the human operator.

5.4.1 Learning Results

The first step is to collect demonstrations. For that, an expert user uses an omega.3 haptic device to teleoperate a 7 DOF Kuka LWR, serving as our remote robot, with force feedback activated, to draw an S-Shape , and to wipe according to an M-Shape, 5 times while making sure that the demonstrations are close to one another. Figures 5.3(a) and 5.3(f) show the used robot setup for both tasks. Then, the obtained raw data is filtered, segmented and re-sampled in order to have data sets of the same length before applying our learning procedure explained in section 5.2. We used $\boldsymbol{K}_c = diag(200, 200, 90)$, a window length $L = 3$ for the stiffness regression, a GMM with 8 states for learning the attractor profile, and with 6 states for learning the stiffness adaptation policy. The number of GMM states was empirically chosen, but methods such as the Bayesian-Information-Criterion (BIC) could be used to find the optimal number of models. As Fig. 5.3(b) and 5.3(g) show, the learnt attractor path via GMM encodes well the acquired demonstrations. Fig. 5.3(d) and 5.3(i) show an example of the $k_{d,t}$ obtained via GMR (solid black), compared to the stiffness profile extracted via regression from equation 5.3 (dotted red). The reconstructed stiffness matrices $\boldsymbol{K}_{r,t}$ are visualized in 2D in Fig. 5.3(e) and 5.3(j) as ellipsoids along the attractor path, where the adaptation is aligned with the current velocity direction. The extracted stiffness matrices are also used to reconstruct the learnt force as the z component of the spring force $\boldsymbol{K}_{r,t}(\hat{\boldsymbol{x}}_{t,d} - \boldsymbol{x}_t)$, which as Fig. 5.3(c), 5.3(g) show, mimics the collected force demonstrations.

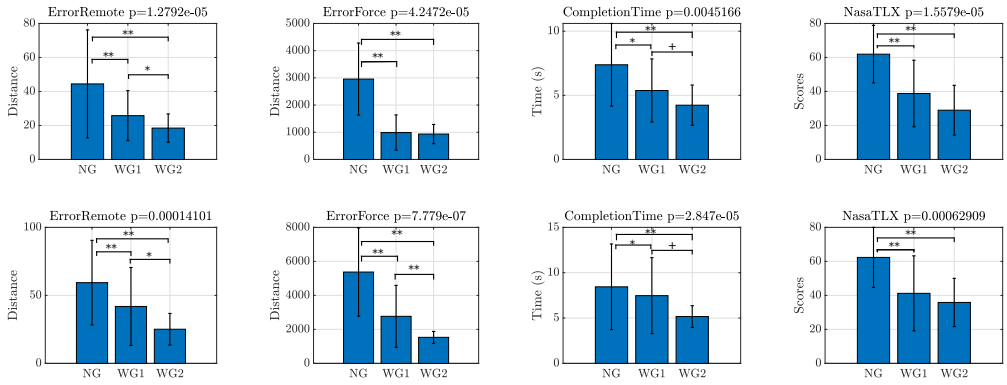


Fig. 5.4: Results of the user study as bar plots, with the upper row as the results of the drawing, while the lower one are the wiping results. The bars show the mean, while the vertical black lines show the standard deviation. The title of each graph shows the computed $p-$ value for the corresponding metric. To indicate significance between conditions, '**' represents $p < 0.01$, '*' represents $p < 0.05$, '+' represents $p < 0.1$. ©2023 IEEE [19]

5.4.2 User Study

Methods

In order to further validate our shared control approach, we conducted a user study where we asked 15 participants to perform drawing on acrylic paper, and wiping on a wooden plate, via teleoperation where we tested three conditions:

- Teleoperation without guidance, with $\boldsymbol{F}_g = \boldsymbol{0}$, $\boldsymbol{W}_e = \boldsymbol{I}_3$ and $\boldsymbol{x}_{r,d} = \boldsymbol{x}_m$, referred to as NG in the following
- Teleoperation with Guidance mode 1, referred to as WG1
- Teleoperation with Guidance mode 2, referred to as WG2

For all conditions, stiffness adaptation via GMR was activated. The experimental procedure was explained to the subjects, in which they were requested to use the haptic device in order to control the KUKA LWR, in order to follow a thick trace of the S-Shape (similar to the one shown in 5.3(a)) in the case of drawing, and the M-shape during wiping. The subjects could clearly see the robot motion and the trace from the position where they stood to perform the teleoperation. Right next to the robot, the subjects were shown on a screen a graph with the actual position of the haptic device in the $y - z$ axis represented as a black dot, and the desired path they should follow as a green line. The path represents the mapping of the learnt force profile (Fig. 5.3(c), 5.3(h)) into a desired motion on the master device. Subjects were instructed to attempt controlling the black dot to follow the green path via the vertical movement of the haptic device handle, while simultaneously controlling the robot motion in the $x - y$ axis to draw the S/M-shape. Since the initial position of the robot was always fixed to have the TCP right on top of the surface, following the trace on the screen via the haptic device motion would lead to

reproducing the desired force profile on the robot, as explained earlier. The graph was only displayed during the NG and WG1 conditions. Subjects were requested to prioritize accuracy and smooth, continuous motions while performing the task, and as a second priority, aim at being quick. The order of conditions and tasks across subjects was block randomized. Once the first task was finished, subjects were provided 1-2 minutes break during which the experimenter changed the tool of the robot to prepare for the next task. Subjects started by performing two trials per condition for familiarization. The familiarization trials were performed with the task the subjects had to perform first. The actual experiment followed with three trials per condition that were used for data analysis. Following each condition, the subjects were requested to fill the Nasa TLX questionnaire, and were asked the following questions to assess the Guidance Quality (referred to as the GQ questionnaire in the following), similar to [203]:

- Q1: Do you feel the guidance useful?
- Q2: Do you have to fight the guidance?
- Q3: Do you feel in control while being assisted?

In both tasks, we set $\boldsymbol{K}_c$ similar to the previous section. For the haptic guidance (5.6), we experimentally set the first eigenvalue of $\boldsymbol{D}_f$ to 70 while the remaining ones to 50. We used $\boldsymbol{K}_g = 330\boldsymbol{I}_3$ and $\boldsymbol{D}_g = 24\boldsymbol{I}_3$ while the thresholds for α were set $e_{max} = 0.05$m and $e_{min} = 0.01$m.

Data Analysis

To evaluate performance, we consider 4 objective metrics: 1) The error between the actual remote robot motion in the $x-y$ plane and the desired one, as well the 2) error between the actual and desired remote robot forces in the $z-$ axis. Error between the profiles was computed via Dynamic Time Warping, which computes a distance as a similarity measure between two time series possibly varying in speed and length [243]. Furthermore, we computed the 3) Root-Mean-Square (RMS) of the master motion Jerk in all three axis, and finally 4) the completion time. For the subjective responses, we consider the weighted scores of the Nasa TLX, as well as the responses of the GQ questionnaire. Then, we compute the mean across trials and subjects for all metrics. For statistical analysis, we use Shapiro-Wilk test to test for normality. If data is normally distributed, we use the repeated-measures Annova followed by pairwise t-tests, and otherwise the Friedemann test, followed by the Wilcoxon signed-rank test for pairwise comparisons. Subsequently, p-values were subjected to Bonferroni corrections, where the alpha value for significance is set to 0.05.

Results

The results of 4 metrics are depicted in Fig. 5.4, as bar plots that show the mean across trials and subjects for each considered metric, as well as the standard deviation,

where the upper row shows the results for the drawing, while the lower one is the wiping. For brevity, we only report the statistically significant results, that are not displayed in the plots: Master motion jerk was decreased for both WG1 ($p = 0.054199$) and WG2 ($p = 0.016$) compared to NG in the case of drawing, and also during wiping ($p < 0.001$ for both WG1 and WG2). For the GQ questionnaire, in drawing, significant higher scores for both WG1 and WG2 were obtained compared to NG for the Q1 and Q3, related to the guidance usefulness and feeling in control during the guidance. For the GQ questionnaire from the wiping, a similar result was achieved, with the difference that a significant higher score was achieved for WG2 compared to WG1.

5.5 Discussion

The results of our user study serve as a proof-of-concept of our designed shared control framework to improve performance for contact tasks execution, which is also in line with previous literature in the shared control domain e.g. [209]. Providing haptic cues helps increase the situation awareness of the human guiding him/her towards a safe and accurate task execution, while alleviating some of the difficulties typically faced in teleoperation in terms of the cognitive load needed to control a robotic device with multiple DOF, as well as performing necessary transformations and mappings between the master and remote robots.
Furthermore, for tasks that require the remote robot to perform contact with the remote environment, reflecting the force sensed by remote robot to the master interface is crucial in providing the human with a sense of telepresence in order to "feel" the environment. This however increases the difficulty of teleoperation due to well-known problems such as the induced master motion [238] where involuntarily human movements resulting from the force feedback create an unstable internal loop that causes oscillations and makes it difficult to maintain a smooth contact. We showed in the previous chapter that an AA that adapts automatically the stiffness of the remote helps reducing this problem, by favoring low stiffness teleoperation whenever possible. This corresponds to our no-guidance condition where the stiffness adaptation happens autonomously, while the human is in charge of commanding the remote robot motion. Adding haptic guidance further assists the human in overcoming these effects and improves task performance. This was also reflected by lower jerk results with guidance activated.
Additionally, a large variety of contact tasks require the hybrid application of specific motion and force profiles, often in decoupled subspaces e.g. drawing, polishing. This implies the requirement for the human to focus on performing simultaneously two unrelated task aspects which further increases the mental demand. We noticed during the teleoperation when guidance is deactivated that many participants either solely focus on following the desired shape while neglecting the application of the force profile on the screen, or alternatively divide the task into sequences where they perform a part of the motion, followed by a stop in the $x-y$ plane such that they can control the haptic device movement in the $z-$ plane to regulate the forces. Haptic

guidance allowed for a smooth, natural and more accurate task execution, reflected by lower motion and force errors, as well as lower execution times.
Our proposed shared control approach essentially implements virtual fixtures that guide the human towards a path that captures the task dynamics and encodes the demonstrated motion and force profiles. The first guidance mode can be easily categorized as a form of Haptic Shared Control [203] where forces exerted by the human, guidance forces and sensed external forces are fused in all the controlled DOF at the haptic device interface, the output of which serves as reference to the remote manipulator. The second mode is a Partitioned Space Architecture, that intuitively exploits information about the task frame to make the human in charge of the motion controlled directions, while the AA regulates the forces to the desired values, thereby further reducing the mental demand of the human operator. In the same manner, guidance is provided in the motion-controlled direction, while the environment feedback force is provided in the force controlled subspace. The results of our user study clearly showed a significant improvement for WG2 compared to WG1 in several metrics, which can be attributed to the lower mental and physical demands on the operator to control only the motion, as well as the reduced force feedback received from the environment .
The presented pipeline in this work can be adequately deployed for the learning and subsequent guidance generation of a large variety of contact tasks. It should be noted however that we assume that the task conditions during demonstrations will remain the same during execution, which clearly lacks the generalization capability. To that end, our framework can be integrated within a higher level of intelligence that uses task-parameterized learning to fit the learning models depending on the inferred task conditions. Alternatively, approaches such as [218] can be exploited to adapt the learnt task models on the fly, whenever needed.

5.6 Summary

In this chapter, we proposed a shared control framework for contact tasks. Our approach exploits learning from demonstration due to its flexibility and generalization capability in modeling a task. The learnt model encodes all the information needed by a robot performing contact, such as the desired motion, forces and task frames in a single entity called the attractor path, which is then used as a reference by a VF to guide the human operator. The guidance forces are generated while taking into account the current human state, based on a control law that attempts to keep the user close to the path, while also providing assistance to move forward along the desired motion. In addition to that, the learnt model autonomously adapts the remote robot stiffness as dictated by the environment state. Furthermore, we presented a user study to validate our proposed shared control approach.

CHAPTER 6

Variable Stiffness Control of Dynamical Systems

In this chapter, we present an approach to encode variable stiffness behaviors into Dynamical Systems (DS), controlled in a closed-loop configuration. Given a desired robot motion represented as a first-order DS, as well as a desired stiffness profile, our approach generates a controller, called Variable Stiffness DS (VSDS), that modulates the robot motion and impedance simultaneously. The robot shows a symmetric spring-like attraction behavior around a reference path described by one of the integral curves of the DS with an interactive behavior prescribed by the desired stiffness. Our approach is validated in simulations and in real robot experiments, displaying a safe and compliant interaction in the face of disturbances or possible collisions, and in contact tasks that requires continuously varying stiffness levels. Additionally, we showcase an application for VSDS in a shared control scenario. In particular, we present a framework that deploys first-order Dynamical Systems (DS) as motion generators providing the desired reference motion, and VSDS for haptic guidance. We show how to shape several features of our controller in order to achieve authority allocation, local motion refinement, in addition to the inherent ability of the controller to automatically synchronize with the human state during joint task execution. Subsequently, we propose a modified formulation of VSDS to further improve its safety and performance. First, we present two solutions to enable a robot to follow a desired velocity profile, in a manner similar to trajectory tracking controllers, while maintaining the closed-loop configuration. Second, we exploit the concept of energy tanks in order to guarantee the passivity during interactions with the environment, as well as the asymptotic stability in free motion, of our closed-loop system. Finally, we show how to extend the VSDS concept to the orientation case based on Unit Quaternions (UQ). We rely on tools from Lie theory to formulate our algorithm, since unlike positions, UQ feature constraints that should be respected in the devised controller. The contents of this chapter are based on the following publications [20, 21, 23, 24]

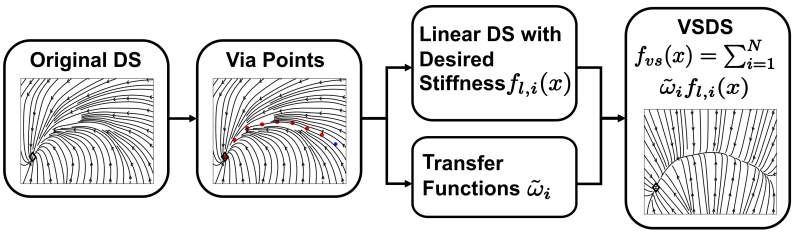


Fig. 6.1: Approach overview of generating Variable Stiffness Dynamical Systems (VSDS).

- *X. Chen*, Y. Michel * and D. Lee, "Closed-Loop Variable Stiffness Control of Dynamical Systems," IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids), pp. 163-169, 2021.*

- *H. Xue*, Y. Michel *, and D. Lee, "A shared control approach based on first-order dynamical systems and closed-loop variable stiffness control," Journal of Intelligent and Robotic Systems, vol. 109, no. 4, pp. 85-98, 2023.*

- *Y. Michel, M. Saveriano, and D. Lee, "A passivity-based approach for variable stiffness control with dynamical systems," IEEE Transactions on Automation Science and Engineering, 2023.*

- *Y. Michel, M. Saveriano, F. J. Abu-Dakka, D. Lee : "Orientation Control with Variable Stiffness Dynamical Systems". IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4457-4463, 2023.*

6.1 Motivation

As reviewed earlier in section 2.1.3, the concept of Feedback Motion Planning (FMP) allows to produce reactive motion generators, that enhance the robot adaptation capabilities in dynamic environments. In contrast to classical motion generators (e.g splines) that are agnostic to the actual robot state, FMP leverages motion planners that are continuously updated based on the measured robot, and possibly environment information. One possibility to achieve such reactivity is on the planning level [133–135], where the desired path or trajectory of the robot is updated based on the current state. Alternatively, a different strategy is to have a tight coupling between motion generation and control, such that reactivity is achieved on control level, which is the main focus in this thesis. This is the case for example in the well-known concept of potential fields [136], or in controllers such as [140] and [139] which rely on velocity fields to encode the desired motion.
Closely related to velocity fields, first-order DS motion generators are becoming increasingly popular thanks to their stability features in terms of generating convergent motions to a global attractor regardless of the initial position. This is in addition to

*Equal Contribution

their flexibility in modeling a myriad of robotic tasks [244–248], and their ability to incorporate a wide range of machine learning algorithms, such as Gaussian mixtures models [13, 249] and Gaussian Processes [250, 251]. Furthermore, the DS formulation (being essentially a velocity field) naturally extends to closed-loop configuration control, allowing to fully exploit their inherent reactivity and stability properties. This was initially shown in [131], where a passive controller was developed to track the velocity of a first-order DS by selectively dissipating kinetic energy in directions perpendicular to the desired motion. However, the controller in [131] is flow-based and therefore does not possess the ability to restrict the robot along a desired path. In other words, the controller does not feature the spring-like attraction behavior characteristic of stiffness, where the robot is pulled back to the path following a perturbation. This was remedied in [252, 253], where the proposed control approach is still flow-based, however with the spring-like attraction behavior embedded inside the DS.
In this chapter, we build on the recent body of literature on closed-loop motion generation, DS and variable impedance control. We propose an approach to encode a desired stiffness profile into first-order DS controlled in a closed-loop configuration. Our work aims at further expanding the potential of DS beyond motion tasks, offering the possibility of changing the apparent robot dynamics, which can be crucial in physical interactions. Given a DS that describes the nominal robot motion and a desired stiffness profile, our proposed method generates a new force field that allows the robot to follow the desired motion in a closed-loop manner with an interactive behavior dictated by the desired stiffness (Fig. 6.1). While [254] proposed how to alternate between fully stiff and compliant (gravity compensation) modes, a wide spectrum of stiffness values between the two extremes can be encoded within our approach. Compared to [252, 253], we are able to encode a continuously varying stiffness profile, independently of the number of primitives used to represent the motion[1], and therefore the specification of the stiffness profile becomes completely decoupled from the desired motion

6.2 Closed-Loop Control Approach

6.2.1 Problem Statement

Let $\boldsymbol{\xi} \in \mathbb{R}^m$ be a generalized state variable (robot joint or Cartesian positions) representing m DOF, we consider in the following a gravity compensated robot with dynamics described by:

$$\boldsymbol{M}(\boldsymbol{\xi})\,\ddot{\boldsymbol{\xi}} + \boldsymbol{C}\left(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}\right)\dot{\boldsymbol{\xi}} = \boldsymbol{u}_c + \boldsymbol{u}_e \tag{6.1}$$

where $\boldsymbol{M}(\boldsymbol{\xi}) \in \mathbb{R}^{m\times m}$ corresponds to the Inertia matrix and $\boldsymbol{C}\left(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}\right) \in \mathbb{R}^{m\times m}$ is the Coriolis matrix. The control and external wrenches are denoted $\boldsymbol{u}_c \in \mathbb{R}^m$ and

[1] For instance, a simple linear motion that can be encoded with only one primitive may require rather complex and continuously changing stiffness requirements

$\boldsymbol{u}_e \in \mathbb{R}^m$, respectively.

Given a first order DS $\boldsymbol{f}_g(\boldsymbol{\xi})$ that describes a nominal motion plan, the controller in [131] computes $\boldsymbol{u}_c$ by tracking the desired velocity $\boldsymbol{f}_g(\boldsymbol{\xi})$, formulated as

$$\boldsymbol{u}_c = -\boldsymbol{D}(\boldsymbol{\xi})\left(\dot{\boldsymbol{\xi}} - \boldsymbol{f}_g(\boldsymbol{\xi})\right) \tag{6.2}$$

where $\boldsymbol{D}(\boldsymbol{\xi}) \in \mathbb{R}^{m\times m}$ is a state-varying damping matrix chosen such that:

$$\boldsymbol{D}(\boldsymbol{\xi}) = \boldsymbol{Q}(\boldsymbol{\xi})\boldsymbol{\Lambda}\boldsymbol{Q}(\boldsymbol{\xi})^T \tag{6.3}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$ is a diagonal matrix with $\lambda_1, \cdots, \lambda_m \geq 0$ indicating damping values and $\boldsymbol{Q}(\boldsymbol{\xi}) \in \mathbb{R}^{m\times m}$ is the direction matrix. The columns of $\boldsymbol{Q}(\boldsymbol{\xi})$ form an orthonormal basis, with the first column defined as $\frac{\boldsymbol{f}_g(\boldsymbol{\xi})}{\|\boldsymbol{f}_g(\boldsymbol{\xi})\|}$. By this definition, $\boldsymbol{f}_g(\boldsymbol{\xi})$ is the first eigenvector of $\boldsymbol{D}(\boldsymbol{\xi})$, with corresponding eigenvalue λ_1, and therefore the controller in (6.2) can rewritten as:

$$\boldsymbol{u}_c = \lambda_1 \boldsymbol{f}_g(\boldsymbol{\xi}) - \boldsymbol{D}(\boldsymbol{\xi})\dot{\boldsymbol{\xi}} \quad . \tag{6.4}$$

This can be physically interpreted as injecting energy along the integral curves of $\boldsymbol{f}_g(\boldsymbol{\xi})$, while providing selective dissipation in directions perpendicular to the motion. Combining (6.1) and (6.2), the closed-loop dynamics becomes:

$$\boldsymbol{u}_e = \boldsymbol{M}(\boldsymbol{\xi})\ddot{\boldsymbol{\xi}} + \left(\boldsymbol{D}(\boldsymbol{\xi}) + \boldsymbol{C}\left(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}\right)\right)\dot{\boldsymbol{\xi}} - \lambda_1 \boldsymbol{f}_g(\boldsymbol{\xi}) \tag{6.5}$$

Comparing (6.5) with the classical impedance control formulation (see section 2.2.4) in an open-loop control configuration, $\boldsymbol{M}(\boldsymbol{\xi})$ and $\boldsymbol{D}(\boldsymbol{\xi}) + \boldsymbol{C}\left(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}\right)$ represent the inertia and damping, while $-\lambda_1 \boldsymbol{f}_g(\boldsymbol{\xi})$ can be seen as a non-linear stiffness term whose stiffness around the equilibrium of $\boldsymbol{f}_g(\boldsymbol{\xi})$ is:

$$\boldsymbol{K}(\boldsymbol{\xi}) = -\lambda_1 \frac{\partial \boldsymbol{f}_g(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \tag{6.6}$$

As explained in [131, 253], the current formulation cannot provide a spring-like symmetrical attraction around an arbitrary reference path, described by one of the integral curves of $\boldsymbol{f}_g(\boldsymbol{\xi})$. Furthermore, there exists a strong coupling between the stiffness behavior and the damping due to λ_1. In addition to that, the stiffness behavior is heavily influenced by the characteristics of the DS describing the motion plan, due to the presence of the Jacobian $\frac{\partial \boldsymbol{f}_g(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}$ in (6.6). This makes it difficult to encode a user-desired stiffness. To solve this problem, we propose a new controller that regulates the robot motion based on $\boldsymbol{f}_g(\boldsymbol{x})$, while shaping the stiffness according to the desired profile. Our VSDS formulation is composed of multiple linear DS, where each locally encodes the desired stiffness behavior, around a set of carefully chosen local attractors (Fig. 6.2). The goal of our approach can be summarized as follows:

Assume that a DS describing a desired motion plan $\boldsymbol{f}_g(\boldsymbol{x})$, with the global point attractor $\boldsymbol{x}^*$, and a desired stiffness profile $\boldsymbol{K}_{des}(\boldsymbol{x})$ are given. The goal of the VSDS controller is to follow this motion plan in a closed-loop control configuration, while the interaction behavior of the robot is described by the desired stiffness profile.

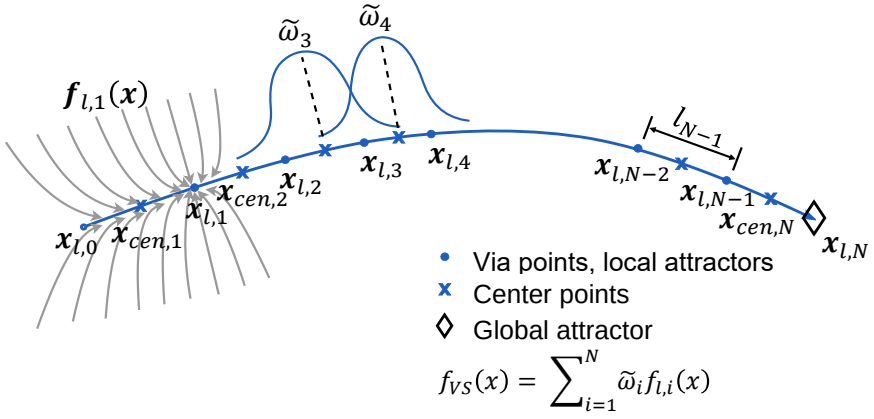


Fig. 6.2: Illustration of the VSDS algorithm.©2021 IEEE [23].

6.2.2 Controller Formulation

In the coming subsections, the key elements of our algorithm are discussed in more details. Without loss of generality, for ease of exposition, we assume that the generalized state $\boldsymbol{\xi}$ describes the 2 dimensional cartesian robot position, such that $\boldsymbol{\xi} = \boldsymbol{x} \in \mathbb{R}^2$.

Sampling Via Points

To construct VSDS, a sequence $\boldsymbol{x}_{sam} = \{\boldsymbol{x}_{l,0}...\boldsymbol{x}_{l,i}...\boldsymbol{x}_{l,N}\}$ of a starting point and N equidistant via points are first generated. The path of the desired DS $\boldsymbol{f}_g(\boldsymbol{x})$ starting from an initial point $\boldsymbol{x}_0$ is first integrated by Euler's method and a temporary point sequence $\boldsymbol{x}_{temp}$ is generated. Then $\boldsymbol{x}_{sam}$ is selected from $\boldsymbol{x}_{temp}$ such that they equally divide the path. These via points shape the motion and act as local attractors of N linear DS. By choosing equidistant points, we are ensuring that the velocity profile is smooth. The sampling process is specified in Algorithm 1, and is encapsulated in function *SampleViaPoints*, Line 2 of Algorithm 3, and further illustrated in Fig. 6.3.

Encoding Stiffness

Although the stiffness in (6.6) is complex with the non-linear DS, the jacobian of a linear DS $\boldsymbol{f}_l(\boldsymbol{x}) = \boldsymbol{A}_l(\boldsymbol{x} - \boldsymbol{x}^*)$ is simply the constant matrix $\boldsymbol{A}_l$ since

$$\frac{\partial \boldsymbol{f}_l(\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{A}_l(\boldsymbol{x} - \boldsymbol{x}^*)}{\partial \boldsymbol{x}} = \boldsymbol{A}_l\,. \tag{6.7}$$

We will exploit this property to encode our desired stiffness into N linear DS by shaping their corresponding Jacobians. With eigenvalue decomposition, we define the constant matrix $\boldsymbol{A}_{l,i}$ of the i_{th} linear DS as the multiplication of a direction matrix and an eigenvalue matrix, such that

$$\boldsymbol{A}_{l,i} = -\boldsymbol{Q}_i \boldsymbol{K}_{des,i} \boldsymbol{Q}_i^T \tag{6.8}$$

Algorithm 1: SampleViaPoints

input : Desired DS $\boldsymbol{f}_g$, robot initial position $\boldsymbol{x}_0$, via point number N
output : $\boldsymbol{x}_{sam}$

$n = 0$;
$k = 1$;
$\Delta t = 0.01$;
$d_{sum} = 0$;
$\boldsymbol{x}_{sam,0} = \boldsymbol{x}_0$;
$\boldsymbol{x}_{temp,k} = \boldsymbol{x}_0$;
while $\|\boldsymbol{x}_{temp,k} - \boldsymbol{x}^* > \epsilon\|$ **do**
    $k = k + 1$;
    $\boldsymbol{x}_{temp,k} = \boldsymbol{x}_{temp,k-1} + \Delta t \boldsymbol{f}_g\left(\boldsymbol{x}_{temp,k-1}\right)$;
    $d_{sum} = d_{sum} + \|\boldsymbol{x}_{temp,k} - \boldsymbol{x}_{temp,k-1}\|$;
end
$d_l = d_{sum}/N$;
for $i \leftarrow 1$ **to** k **do**
    if $\|\boldsymbol{x}_{temp,i} - \boldsymbol{x}_{sam,n}\| \geq d_l$ **then**
        $n = n + 1$;
        $\boldsymbol{x}_{sam,n} = \boldsymbol{x}_{temp,i}$;
    end
end
$\boldsymbol{x}_{sam,N} = \boldsymbol{x}^*$;

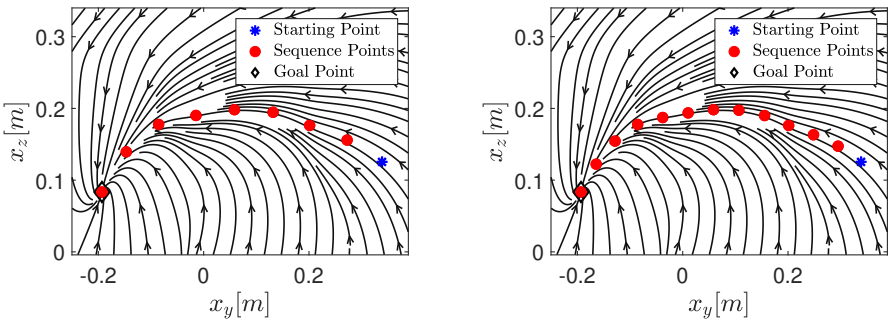


Fig. 6.3: Generated via points with fixed points number N. Left: $N = 8$ and right: $N = 12$. In the background, the streamlines of the desired DS $\boldsymbol{f}_g\left(\boldsymbol{x}\right)$ are shown. ©2021 IEEE [23].

for $i = 1 \cdots N$. The matrix $\boldsymbol{K}_{des,i}$ is a diagonal positive definite matrix that indicates the stiffness at the ith sampled via point. It is defined as

$$\boldsymbol{K}_{des,i} = \boldsymbol{K}_{des}\left(\boldsymbol{x}_{l,i}\right) = \begin{pmatrix} k_{i,1} & 0 \\ 0 & k_{i,2} \end{pmatrix} \tag{6.9}$$

which represents the value of the desired stiffness profile at $\boldsymbol{x}_{l,i}$. The eigen values $k_{i,1}$ and $k_{i,2}$ are both greater than zero, and represent the stiffness along the motion

direction and orthogonal to it, respectively. The direction matrix of the i_{th} linear DS is defined as:

$$\boldsymbol{Q}_i = [\boldsymbol{e}_{i,1}, \boldsymbol{e}_{i,2}] \tag{6.10}$$

where $\boldsymbol{e}_{i,1}$ is parallel to the velocity at state $\boldsymbol{x}_{l,i}$, i.e. $\boldsymbol{e}_{i,1} = \frac{\boldsymbol{f}_g(\boldsymbol{x}_{l,i})}{\|\boldsymbol{f}_g(\boldsymbol{x}_{l,i})\|}$ [2], and $\boldsymbol{e}_{i,2} \perp \boldsymbol{e}_{i,1}$ is orthonormal to $\boldsymbol{e}_{i,1}$. Intuitively, the projection (6.8) interprets the first eigen value of $\boldsymbol{K}_{des}$ to be the stiffness along the direction of motion, while the remaining eigen values as the stiffness perpendicular to the current motion direction. Function *FindOrthonormalBasis* in Line 5 of Algorithm 3 calculates the orthonomal vector of the input vector. We can then define the i_{th} linear DS as

$$\boldsymbol{f}_{l,i}(\boldsymbol{x}) = \boldsymbol{A}_{l,i}(\boldsymbol{x} - \boldsymbol{x}_{l,i}). \tag{6.11}$$

Transfer Functions

As explained earlier, our VSDS is defined as the weighted sum of N local, linear DS. The role of the weights is to act as state-dependent activation throughout the state space and enable the smooth transition between the local DS. To this end, we will start by defining our distance-dependent Gaussian kernels as:

$$\omega_i(\boldsymbol{x}) = \exp\left(-\frac{(\boldsymbol{x} - \boldsymbol{x}_{cen,i})^T(\boldsymbol{x} - \boldsymbol{x}_{cen,i})}{2(\sigma^i)^2}\right) \tag{6.12}$$

where $\boldsymbol{x}_{cen,i} = \frac{1}{2}(\boldsymbol{x}_{l,i} + \boldsymbol{x}_{l,i-1})$ denotes the center of i_{th} linear DS, while $\sigma^i \in \mathbb{R}^+$ is the smoothing parameter that is proportional to the distance between the via points and controls the region of influence of each linear DS, i.e. $\sigma^i = \delta l_i$ for $l_i = \|\boldsymbol{x}_{l,i} - \boldsymbol{x}_{l,i-1}\|$ and $\delta \in \mathbb{R}^+$. We can now define our weighting function as the normalization of these kernels:

$$\tilde{\omega}_i(\boldsymbol{x}) = \frac{\omega_i(\boldsymbol{x})}{\sum_{j=1}^{N}\omega_j(\boldsymbol{x})}. \tag{6.13}$$

Regeneration

With the encoded stiffness, the robot has a behavior of symmetrically returning to the initial reference path, from anywhere in the state space. This implies that the further away the robot is from the path, the larger the control torque needed to attract back the robot to its reference. This might result in sudden robot motions with large accelerations, which pose safety concerns in human-robot interaction or on the robot itself. To avoid that, we modify our VSDS to endow it with a motion level compliance behavior [255]: when a perturbation pushes the robot outside an attraction tube area around its reference path, the robot gives up the current path and generates a new VSDS, with a new associated path online. Such a feature can

[2] we set a threshold on $\|\boldsymbol{f}_g(\boldsymbol{x}_{l,i})\|$, when the value is close to 0, we use the value of the previous via-point i.e $\boldsymbol{f}_g(\boldsymbol{x}_{l,i}) = \boldsymbol{f}_g(\boldsymbol{x}_{l,i-1})$

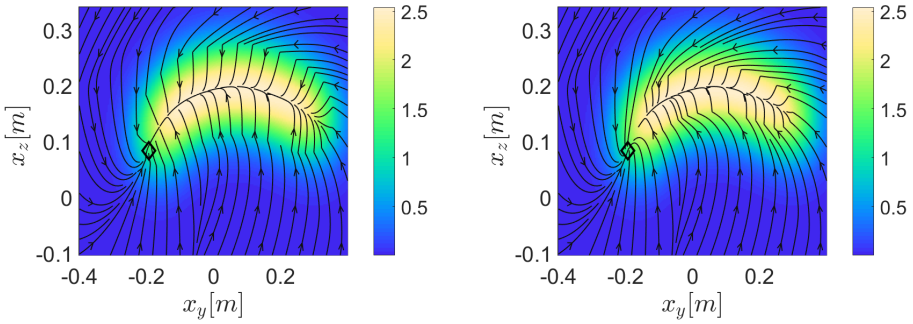


Fig. 6.4: Visualization of the effect of $\bar{\omega}_{th}$. The tube-like area with the symmetric attraction behavior is highlighted in yellow. The left figure shows an example of a large attraction area with $\bar{\omega}_{th} = 1$, compared to a relatively smaller area in the right figure with $\bar{\omega}_{th} = 2$. ©2021 IEEE [23].

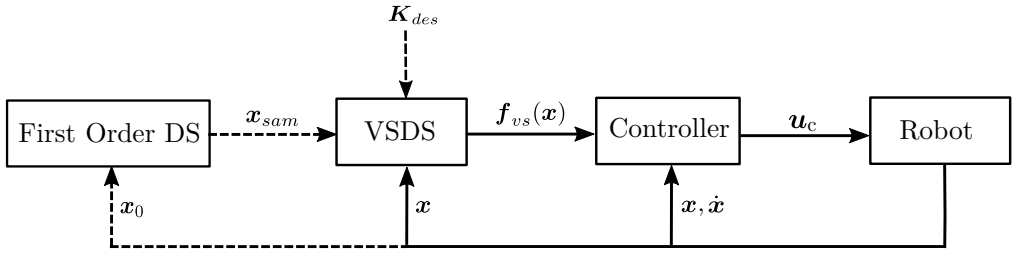


Fig. 6.5: Block diagram showing the VSDS control architecture. The dotted lines mean that the information is used only during the initialization phase.

be also beneficial in interactive teaching or shared control scenarios, as it will be shown in the next section. During execution, a real-time check is applied, where the value $\bar{\omega}(\boldsymbol{x}) = \sum_{i=1}^{N} \omega_i(\boldsymbol{x})$ is compared with a pre-defined threshold value $\bar{\omega}_{th}$. If the current position is outside the local attraction area, the approach restarts by sampling new via points from the DS $\boldsymbol{f}_g(\boldsymbol{x})$, with $\boldsymbol{x}_0 = \boldsymbol{x}$ i.e starting from the current robot position. Fig. 6.4 illustrates the effect of different $\bar{\omega}_{th}$ values on the size of the attraction area. One can choose this value based on the task requirements or the desired stiffness profile.

Controller formulation

Finally, we construct our VSDS that encodes both the motion and the stiffness as the concatenation of the weighted linear local DS. This can be expressed as

$$\boldsymbol{f}_{vs}(\boldsymbol{x}) = \sum_{i=1}^{N} \tilde{\omega}_i(\boldsymbol{x}) \boldsymbol{f}_{l,i}(\boldsymbol{x}) \ . \tag{6.14}$$

The controller signal is then computed as

$$\boldsymbol{u}_c = \boldsymbol{f}_{vs}(\boldsymbol{x}) - \boldsymbol{D}(\boldsymbol{x}) \dot{\boldsymbol{x}} \tag{6.15}$$

Algorithm 2: Robot Control Algorithm with VSDS

input : $\boldsymbol{f}_g$, δ, N and $\boldsymbol{K}_{des}$

Algorithm 3;
while $\|\boldsymbol{x} - \boldsymbol{x}^* > \epsilon\|$ **do**
  for $i \leftarrow 1$ **to** N **do**
    $\omega_i(\boldsymbol{x}) = \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}_{cen,i})^T(\boldsymbol{x}-\boldsymbol{x}_{cen,i})}{2(\sigma^i)^2}\right)$;
  end
  $\bar{\omega}(\boldsymbol{x}) = \sum_{i=1}^{N} \omega_i(\boldsymbol{x})$;
  if $\bar{\omega}(\boldsymbol{x}) < \bar{\omega}_{th}$ **then**
    Algorithm 3;
  end
  for $i \leftarrow 1$ **to** N **do**
    $\boldsymbol{f}_{l,i}(\boldsymbol{x}) = \boldsymbol{A}_{l,i}(\boldsymbol{x} - \boldsymbol{x}_{l,i})$;
    $\tilde{\omega}_i(\boldsymbol{x}) = \frac{\omega_i}{\sum_{j=1}^{N} \omega_j(\boldsymbol{x})}$;
  end
  $\boldsymbol{f}_{vs}(\boldsymbol{x}) = \sum_{i=1}^{N} \tilde{\omega}_i(\boldsymbol{x}) \boldsymbol{f}_{l,i}(\boldsymbol{x})$;
  $\boldsymbol{u}_c = \boldsymbol{f}_{vs}(\boldsymbol{x}) - \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}}$;
end

where $\boldsymbol{D}(\boldsymbol{x})$ takes the same form as in (6.3), providing dissipation along and orthogonal to the current motion direction. The final system configuration is depicted in Fig. 6.5. As explained in algorithm 2, the global DS $\boldsymbol{f}_g$ is used to generate a set a of via points that describe the current nominal motion path, which is then utilized to construct the local DS that encodes the desired stiffness. This procedure occurs at the beginning of task execution, or whenever the current position lies outside the local attraction area, around the current reference path. Once the VSDS is constructed, it is used to generate the control signal $\boldsymbol{u}_c$ which is then commanded to the robot actuators.

Regarding the stiffness of the closed-loop system, an important observation in (6.15) is that the stiffness is now completely decoupled from damping. This provides more freedom in shaping the robot impedance, which is typically defined by the robot stiffness and damping. Furthermore, we are able to freely encode a desired constant/state varying stiffness profile that might reflect certain task requirements, regarding the interaction of the robot with its environment. This is highlighted in Fig. 6.6, where depicted are the desired stiffness profiles, compared to the actual stiffness of the system, described by $-\sum_{i=1}^{N} \tilde{\omega}_i(\boldsymbol{x})\boldsymbol{A}_{l,i}$. Clearly, our approach is able to track the desired stiffness fairly accurately.

Algorithm 3: Generation of VSDS parameters

Result: Parameters of VSDS

$\boldsymbol{x}_0 = \boldsymbol{x}$;

$\boldsymbol{x}_{sam} = \{\boldsymbol{x}_{l,0}, \boldsymbol{x}_{l,i}, \ldots, \boldsymbol{x}_{l,N}\} =$ `SampleViaPoints(`$\boldsymbol{f}_g, \boldsymbol{x}_0, N$`)`;

for $i \leftarrow 1$ **to** N **do**

  $\boldsymbol{e}_{i,1} = \frac{\boldsymbol{f}_g(\boldsymbol{x}_{l,i})}{\|\boldsymbol{f}_g(\boldsymbol{x}_{l,i})\|}$;

  $\boldsymbol{e}_{i,2} =$ `FindOrthonormalBasis(`$\boldsymbol{e}_{i,1}$`)`;

  $\boldsymbol{K}_{des,i} = \boldsymbol{K}_{des}(\boldsymbol{x}_{l,i})$;

  $\boldsymbol{Q}_i = [\boldsymbol{e}_{i,1}, \boldsymbol{e}_{i,2}]$;

  $\boldsymbol{A}_{l,i} = -\boldsymbol{Q}_i \boldsymbol{K}_{des,i} \boldsymbol{Q}_i^T$;

  $\boldsymbol{x}_{cen,i} = \frac{1}{2}(\boldsymbol{x}_{l,i} + \boldsymbol{x}_{l,i-1})$;

  $l_i = \|\boldsymbol{x}_{l,i} - \boldsymbol{x}_{l,i-1}\|$;

  $\sigma_i = \delta l_i$;

end

6.2.3 Evaluation

In this section, we seek to validate our approach in terms of safety and successful task execution, in real robot experiments. We implemented our algorithm on a 7 DOF KUKA LWR. The algorithm was implemented in C++ on a standard core i7 PC, which communicated with the robot using the Fast Research Interface library. To construct our VSDS, we used the Stable-Estimator-of-Dynamical-Systems (SEDS) [13] to generate the first order DS $\boldsymbol{f}_g(\boldsymbol{x})$ describing the nominal motion plan (see sec. 2.2.5), which was learned from demonstrated data obtained by kineasthetically teaching the robot in gravity compensation mode.

Symmetrical Attraction and Regeneration

In this experiment, we aim to demonstrate the symmetric attraction behavior and the motion level compliance property in our proposed approach. We compare our VSDS to the DS controller in [131] described in 6.2.1, when following the same motion plan generated by a first order SEDS and starting from the same initial position. For the DS controller, we use $\lambda_1 = 150$ and $\lambda_2 = 230$, while for our VSDS we set a constant stiffness $\boldsymbol{K}_{des} = diag\,(300, 500)$ and damping values $\lambda_1 = \lambda_2 = 30$. Fig 6.7 shows the results of the two controllers in response to perturbations. For the DS controller [131], the robot cannot follow the reference path anymore when the perturbations occur at $x_y \approx 0.2$ and $x_y \approx -0.1$. Instead, the controller find a new reference path that follows the integral curves of the nominal SEDS to reach the attractor. On the other hand, our controller is able to stick to its reference path even in the presence of various perturbations at different points across the state space. In the second experiment, we tested the behavior of our controller when subject to relatively larger perturbations, applied by a human directly interacting with the robot. The left figure of Fig. 6.8 shows that, if the perturbation is small, the robot

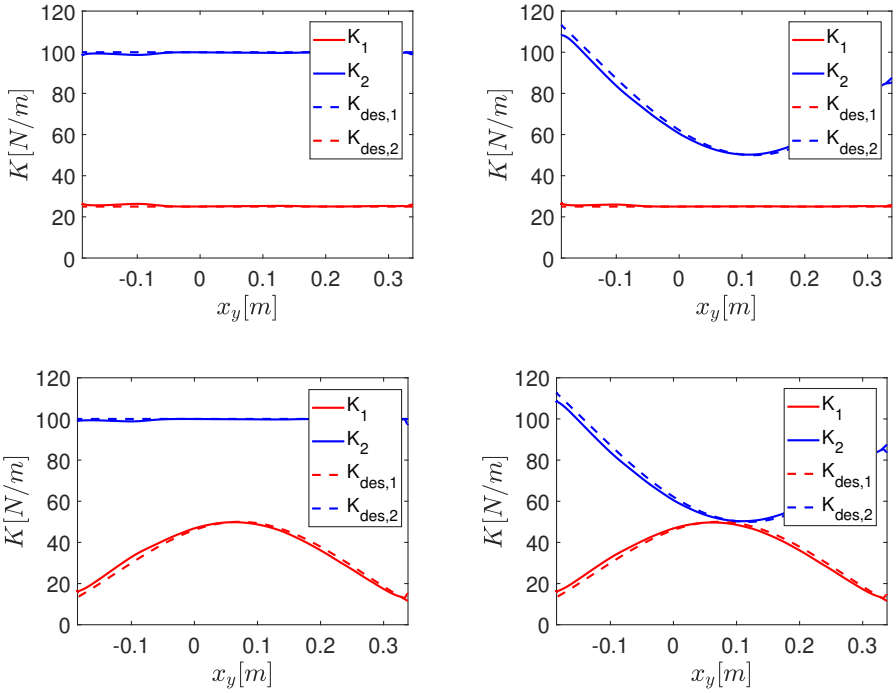


Fig. 6.6: Simulation results showing actual (solid lines) vs. desired (dotted lines) stiffness profiles, with k_1 is the stiffness along the motion, and k_2 is the stiffness perpendicular to it. ©2021 IEEE [23].

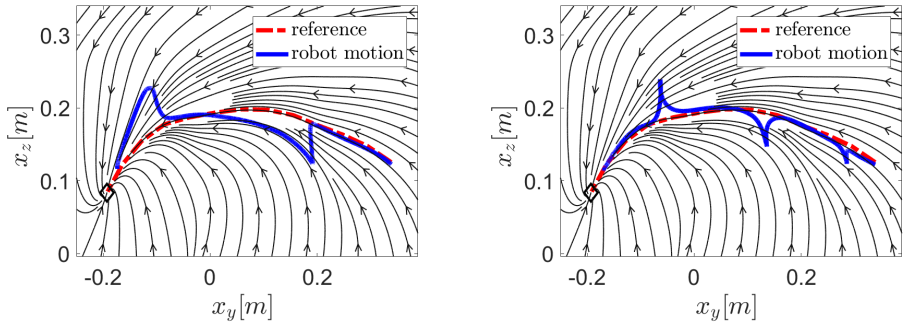


Fig. 6.7: Comparison of motions under perturbations between Motion of SEDS with [131] (left) and Motion of VSDS (right). The streamlines in the background are the learned SEDS. ©2021 IEEE [23].

will go back to the reference path. But if the human applies a perturbation large enough to push the robot far away from its reference path, the robot generates a new path to reach the final goal point. If a small perturbation happens again after the regeneration process, the robot shows the symmetric attraction behavior once more, going back to the new reference path (the right figure of Fig. 6.8).

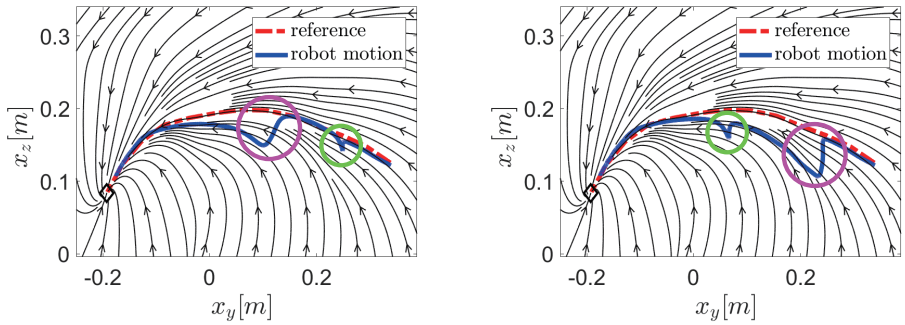


Fig. 6.8: Path regeneration after large perturbation. As robot moves from right to left, the robot returns to a generated path under small perturbations (green circles), while generates a new path based on the motion plan DS under large perturbations (violet circles).©2021 IEEE [23].

Collision Reaction

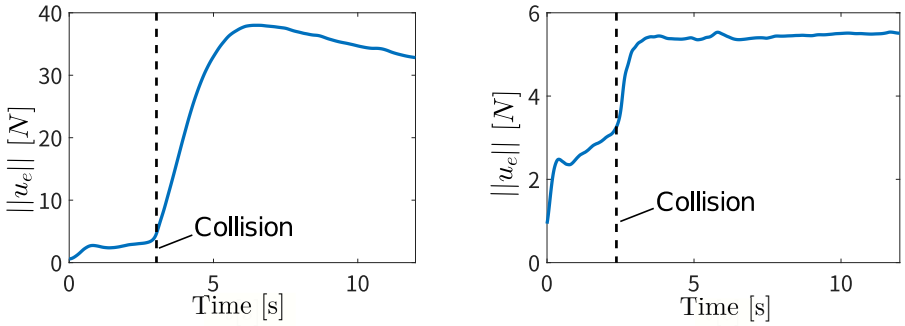
One important advantage of our approach is that it works in a closed-loop configuration. This means that the notion of tracking a time-indexed trajectory does not exist anymore, instead motion generation and control are combined in one loop that is always aware of the current robot state. The aim of the following experiment is to highlight this crucial feature in more depth, in a situation where the actual robot environment might not match the planned scenario e.g a collision.

The setup of our collision reaction experiment is a 7 DOF KUKA LWR and a crash test dummy (Fig. 6.9). The dummy is positioned in such a way that would induce a collision while the robot is executing the desired motion, encoded through SEDS. We compare our VSDS controller with a classical impedance controller tracking a trajectory generated by integrating SEDS in open-loop. For both controllers, we use the same stiffness and damping values. As can be seen in left column of Fig. 6.9, for the classical open-loop impedance controller, the robot keeps pushing the head of the dummy, leading to a contact force that reaches $38N$ (Fig. 6.9(c)). This is due to the accumulated error in the spring term that leads to an increasing control force, that can eventually damage the motors, or a large abrupt acceleration if the obstacle is suddenly released. On the other hand, our controller is clearly very safe in this regard. The contact force remains low at $5.4N$ (Fig. 6.9(d)), and reaches a constant level after the initial collision. It is also worth noting that the controller does not use any force measurements to detect the collision, as done in previous works e.g [256].

Charger Insertion

In this section, we test the ability of our approach to perform a contact task, that requires varying stiffness levels. We chose the task of inserting a charger fixed to the robot end-effector into a socket (Fig. 6.10). For this task, the robot starts with a moderate stiffness, and as it approaches the socket, the stiffness in the direction along the motion k_1 should increase in order to generate a high enough force during the

(a) Snapshot-Open Loop (b) Snapshot-VSDS



(c) External Force-Open Loop (d) External Forces-VSDS

Fig. 6.9: Collision Experiment. The upper row is snapshots after the collision, the red dotted lines mark the collision start position and serve to indicate how much the robot moved after the collision happens. The lower row is the sensed contact forces at the robot end-effector. Left column: SEDS controlled in open-loop, while right column: VSDS. ©2021 IEEE [23].

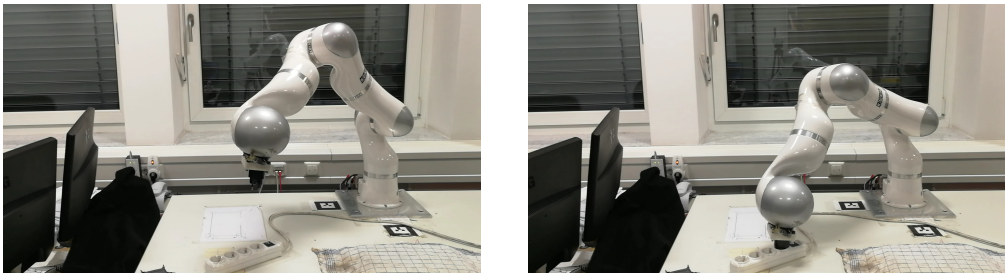
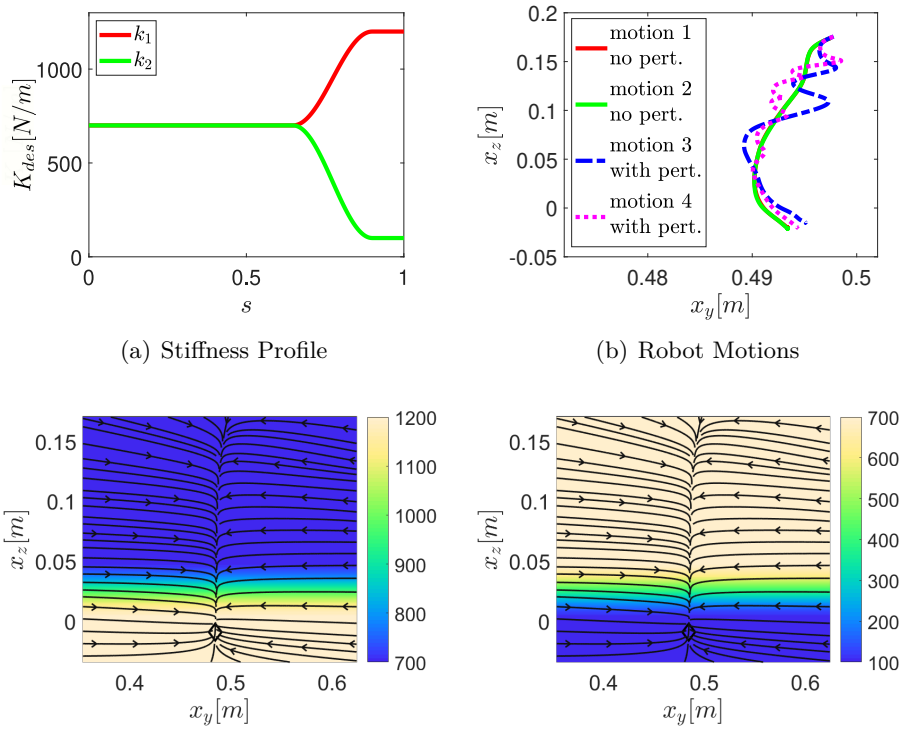


Fig. 6.10: Experimental setup of the charger insertion insertion. Left: Robot at the initial position and Right: After successfully inserting the charger . ©2021 IEEE [23].

insertion phase, while the stiffness perpendicular to motion k_2 should decrease, as the socket slots provide the necessary constraints for constraining the robot motion. A hand crafted state-varying stiffness profile was designed to reflect these requirements (Fig. 6.11(a)) and was encoded in our VSDS. Alternatively, the desired stiffness profile can also be learnt from human demonstration [10] or via reinforcement learning [257] by minimizing a suitable cost function (e.g contact forces during the insertion phase). As for the desired motion, it was provided by a SEDS learned through a one human demonstration. As shown in Fig. 6.11(b), the robot is able to successfully perform the



(a) Stiffness Profile

(b) Robot Motions

(c) Stiffness Color Map in z

(d) Stiffness Color Map in y

Fig. 6.11: Results of the charger insertion task. The upper-right figure shows actual robot motions from different trials with and without perturbations, where all motions accomplish the task. The upper-left figure shows the desired stiffness profile along the path parameter s, which indicates the normalized distance between the start and end points in the $z-$ direction, and where the stiffness starts to increase in the neighborhood of the socket at $s = 0.65$. The figures in lower row show the stiffness along (k_1, left) and perpendicular (k_2, right) to the motion, respectively, as a color map. ©2021 IEEE [23].

task, even in the presence of various perturbations applied at its end-effector. This becomes possible, thanks to the symmetric attraction behavior encoded in our VSDS, enabling the robot to stick to the demonstrated reference path, that successfully achieves the task. Otherwise, in response to perturbations, the robot would attempt to find a new motion plan, which can lead to the robot wrongly approaching the socket (e.g from the sides), resulting in task failure.

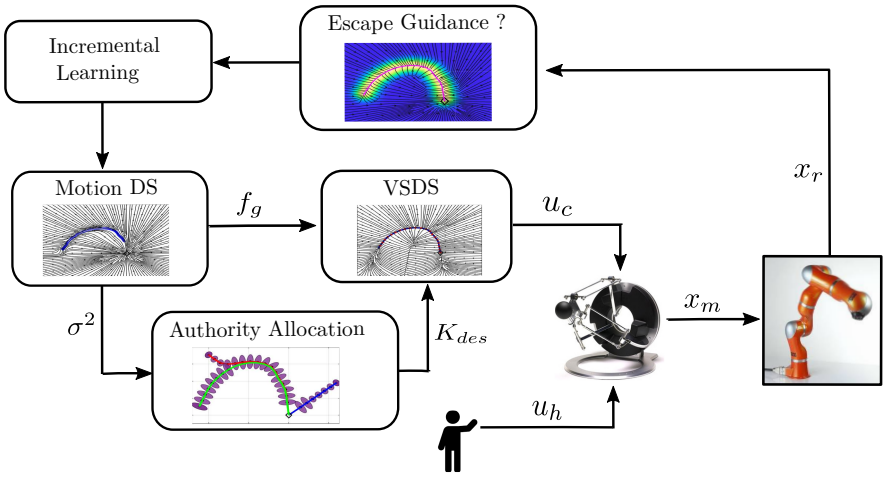


Fig. 6.12: The overall architecture of the proposed shared control approach.

6.3 VSDS for Shared Control

In this section, we present a new shared control architecture that builds on the use of first-order DS as motion generators, and control in closed-loop to generate haptic guidance, where the VSDS controller is exploited. While in the previous section we demonstrated the benefits of our controller for autonomous task execution, in this section, we show how to exploit and adapt our controller features to develop a new shared control approach. For instance, it can be used with any DS, which offers the flexibility to benefit from existing learning/regression techniques available for DS in the literature. The controller is in closed-loop, and therefore synchronizes automatically with the human state. In addition to that, the ability to encode variable stiffness profiles can be used to adjust the strength of the guidance depending on the human confidence or the model knowledge. Moreover, the symmetric attraction behavior means the user is always pulled to a desired path, which can be crucial to successful task execution, in addition to convergence to the global attractor. Finally, this attraction only holds locally, which means that the width of the attraction region can be adjusted to be consistent with the stiffness, and therefore can be designed such that the human can escape the guidance, when needed.

6.3.1 System Architecture

In this section, we consider a teleoperation scenario where a human physically interacts with a master robot with coordinates $\boldsymbol{x}_m$ to control the motion of a remote manipulator with position $\boldsymbol{x}_r$, to complete a desired task. The results however can be straightforwardly extended to the case where the human directly interacts with a robot e.g. in a cooperative manipulation scenario. In the following, we present the fundamental building blocks of our shared control architecture, illustrated in Fig. 6.12. For a complete shared control solution, such a framework would consist of a motion generator that outputs a desired motion plan, and naturally a controller

that provides haptic guidance depending on the desired motion. Furthermore, the strength of this guidance should be adjusted according to some criteria in such a way the authority is arbitrated between the human and the autonomous agent. Finally, the framework should provide an option to the human to locally adapt generated motions depending on changes in the environment or task scenario.

Motion Generation

The first part of the proposed framework is the motion generator, which outputs the desired path for a specific task. Similar to before, this is provided by a first-order time invariant DS. While in principle any state-of-the-art DS approach can be used, in this section, we chose a DS based on the formulation proposed in [250], since it can be seamlessly extended with incremental learning. To learn a DS model, we deploy LfD. Consider the following DS

$$\dot{\boldsymbol{x}}_{d,o} = \boldsymbol{f}_o(\boldsymbol{x}_r) \tag{6.16}$$

where $\boldsymbol{f}_o$ represents a linear globally asymptotically stable DS, while $\dot{\boldsymbol{x}}_{d,o}$ is the desired velocity. Obviously, the velocity of demonstrations will be different from the velocity field described by $\boldsymbol{f}_o$. Through rotating and scaling however, it is possible to reshape $\boldsymbol{f}_o$ to match the demonstrated velocity field. Therefore, LfD becomes the task of learning to reshape the linear DS based on demonstrations. The rotation and scaling parameters can be combined together to form a modulation field $\boldsymbol{T}(\boldsymbol{x}_r)$

$$\boldsymbol{T}(\boldsymbol{x}_r) = (1 + \chi(\boldsymbol{x}_r))\boldsymbol{R}(\boldsymbol{x}_r) \tag{6.17}$$

where $\chi(\boldsymbol{x}_r)$ is the scaling factor, and $\boldsymbol{R}(\boldsymbol{x}_r)$ is the rotation matrix. The rotation matrix has the following form in two-dimensional space

$$\boldsymbol{R}(\boldsymbol{x}_r) = \begin{bmatrix} cos(\phi(\boldsymbol{x}_r)) & -sin(\phi(\boldsymbol{x}_r)) \\ sin(\phi(\boldsymbol{x}_r)) & cos(\phi(\boldsymbol{x}_r)) \end{bmatrix} \tag{6.18}$$

where $\phi(\boldsymbol{x}_r)$ represents the state-dependent rotation angle. The reshaped DS is then expressed as

$$\dot{\boldsymbol{x}}_d = \boldsymbol{f}_g(\boldsymbol{x}_r) = \boldsymbol{T}(\boldsymbol{x}_r)\boldsymbol{f}_o(\boldsymbol{x}_r), \tag{6.19}$$

and does not lead to any spurious attractors or cause divergent behaviors [250]. Learning the reshaped DS from demonstrations is equivalent to learning the state dependent parameters $\phi(\boldsymbol{x}_r)$ and $\chi(\boldsymbol{x}_r)$, termed modulation parameters. The raw collected demonstration data consisting of position and velocity data can be converted to position and modulation parameters, where position data are inputs and modulation parameters are outputs. The detailed conversion process is explained in [250]. We can now use Gaussian Process (GP) to fit the training data, because it enables incremental learning by simply enlarging the training dataset. The squared exponential covariance function between two positions $\boldsymbol{x}$ and $\boldsymbol{x}^{'}$

$$k(\boldsymbol{x}, \boldsymbol{x}^{'}) = \gamma_f \, \exp(-\frac{(\boldsymbol{x} - \boldsymbol{x}^{'})^T(\boldsymbol{x} - \boldsymbol{x}^{'})}{2l}) \tag{6.20}$$

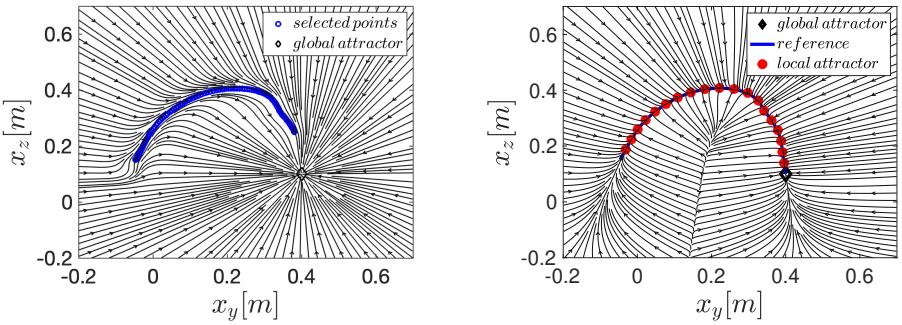


Fig. 6.13: Left: Streamlines of the locally reshaped DS $\boldsymbol{f}_g$ around demonstration data points shown in blue. Right: Streamlines of VSDS symmetrically attracting around a reference path shown in blue, and the red points are local attractors of VSDS. The rhombus in both plots is the global attractor $\boldsymbol{x}^*$

is chosen to construct the covariance matrix, where $\gamma_f, l > 0$ are hyperparameters. Additionally a random Gaussian noise is added in the covariance matrix. In this work, we set the hyperparameters to pre-fixed values.

After fitting the training dataset into the GP model, we use Gaussian Process Regression (GPR) to compute the predicted modulation parameters $\phi(\boldsymbol{x}_r)$ and $\chi(\boldsymbol{x}_r)$, given a certain position $\boldsymbol{x}_r$. GPR outputs a predictive mean value $\boldsymbol{\mu}(\boldsymbol{x}_r)$ and a predictive variance $\sigma^2(\boldsymbol{x}_r)$, which is computed by following the standard expression in GPR [250]. Finally, we obtain the reshaped DS as (6.19). This reshaped DS outputs a motion plan to the global attractor given any starting position. An example of this DS is shown in Fig. 6.13, left.

Haptic guidance

Once the DS is learnt, a controller is needed to provide haptic guidance along the desired motion. This is rendered on the master device, as commonly done with virtual fixtures in the shared control literature [258]. The DS model, however, represents a motion on the remote manipulator side, where the task goal is expressed. To solve this problem, given a desired cartesian position $\boldsymbol{x}_{k,r}$ or velocity $\dot{\boldsymbol{x}}_{k,r}$ on the remote robot side, we map it first to the master side via a suitable coordinate transformation that takes into account workspace differences. Subsequently, the haptic guidance force on the master robot is computed using VSDS as $\boldsymbol{u}_c = \boldsymbol{f}_{vs}(\boldsymbol{x}_m) - \boldsymbol{D}(\boldsymbol{x}_m)\dot{\boldsymbol{x}}_m$, based on the motion plan DS $\boldsymbol{f}_g(\boldsymbol{x}_r)$ transformed on the master side.

Authority Allocation

Another important aspect in shared control is authority allocation. In this work, this is realized by adjusting the strength of the guidance forces. While several metrics can be used, here we show how the commonly used idea, where authority allocation is variance-based (e.g. [208]), can be integrated in our framework. Since the GPR outputs the prediction with a mean and a variance $\sigma^2(\boldsymbol{x}_r)$, we use this variance

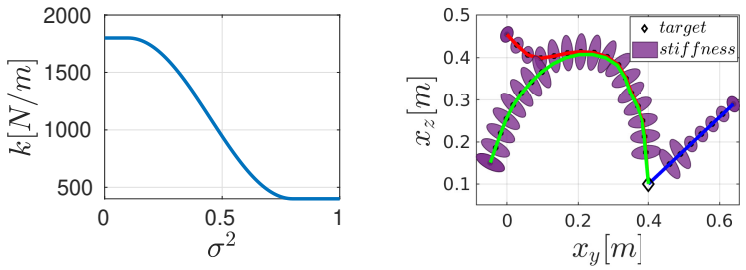


Fig. 6.14: Left: An example plot to show how stiffness changes according to variance. Right: the stiffness along the path shown as ellipses, where wide ellipse means a high stiffness. The paths generated by the reshaped DS are shown in the right figure, where the green path is the demonstrated one, while the blue and red paths are obtained starting from two different positions

information to set the stiffness of our VSDS. We set a high stiffness in regions having low variances, since a low variance output by GPR indicates closeness to demonstrations. This limits the freedom of the human in deviating from the desired motion. Conversely, we set a low stiffness in regions that have high variances which are far from demonstrated motions. This makes it easier for the human to overrule the guidance forces. Therefore, authority allocation is implicitly achieved by adjusting the stiffness.

Taking the i-th local attractor of VSDS as an example, the desired stiffness profile for a planar motion is expressed as

$$\boldsymbol{K}_{des,i} = \begin{pmatrix} k_{i,1} & 0 \\ 0 & k_{i,2} \end{pmatrix} \tag{6.21}$$

where $k_{i,1}$ indicates the strength with which the user is pulled along the trajectory, while $k_{i,2}$ is the stiffness perpendicular to the motion direction and penalizes deviations from the path. We chose to set $k_{i,1}$ to a fixed value, while $k_{i,2}$ is computed according to

$$k_{i,2} = \begin{cases} a_1 + a_2 & \sigma_i^2 < \sigma_l^2 \\ a_1 - a_2 \sin(\dfrac{\pi(\sigma_i^2 - \sigma_l^2)}{\sigma_u^2 - \sigma_l^2} - \dfrac{\pi}{2}) & \sigma_l^2 \leq \sigma_i^2 \leq \sigma_u^2 \\ a_1 - a_2 & \sigma_i^2 > \sigma_u^2 \end{cases} \tag{6.22}$$

where $a_1, a_2, \sigma_l^2, \sigma_u^2$ are predefined thresholds and $\sigma_i^2(\boldsymbol{x}_i)$ is the predictive variance from GPR at the i-th local attractor. The second condition of (12) ensures a smooth transition between the low and high variance states as shown in Fig. 6.14, left.

We illustrate our stiffness setting based on variances in Fig. 6.14, right. The green path is the demonstrated motion, and naturally the variances along this trajectory are very low, resulting in high stiffness values at all the local attractors of VSDS. On the other hand, for the red and blue paths, we can see that the stiffness is low at local attractors far away from the demonstrated trajectory, and increases when the position of the local attractor is closer to or coincides with the demonstrations.

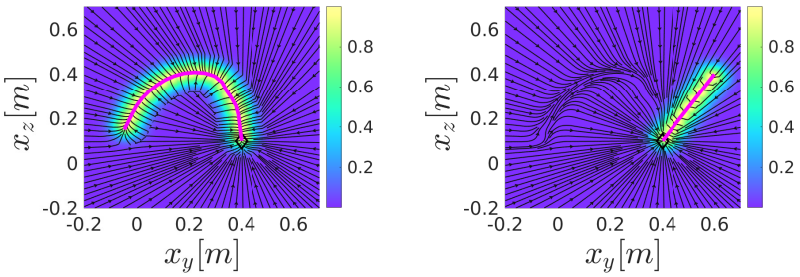


Fig. 6.15: Tunnel region effect of VSDS, where the highlighted area is the region where the symmetric attraction effect is activated, while the purple is the rest of the state space where streamlines follow $\boldsymbol{f}_g$. The left figure shows a relatively wide region, while the region in the right figure is narrower.

Incremental Learning

We complement our shared control architecture with online incremental learning in order to refine learnt motions, or to update task knowledge in regions of the state space not demonstrated before. This implies that the human might need to temporarily escape the guidance, in order to provide new demonstrations. Therefore, we exploit the fact that our VSDS controller can provide local symmetric attraction in a tunnel region around the reference path. When the human operator moves out of the tunnel, no further guidance is applied, and the master interface goes into gravity compensation mode where the human is completely free to manipulate the robot. The new demonstrated path is then used for incremental learning.
As explained in the previous section, the tunnel of VSDS is determined by properly setting the threshold value $\bar{\omega}_{th}$. For each position $\boldsymbol{x}_m$, we check the weights of all local attractors, computed by $\bar{\omega}\left(\boldsymbol{x}_m\right) = \sum_{i=1}^{N} \omega_i\left(\boldsymbol{x}_m\right)$. In this section, we set the threshold value proportionally to the variance of the reference path. First, we sum over the predictive variance from GPR of all attractors along the reference path and compute the average of the variance $\overline{\sigma}^2 = \frac{1}{N}(\sum_{i=1}^{N} \sigma_i^2(\boldsymbol{x}_r))$ where N represents the number of local attractors of VSDS. In this case, the threshold value is state-varying, and is set as

$$\bar{\omega}_{th}(\boldsymbol{x}_m) = \begin{cases} b_1 - b_2 & \overline{\sigma}^2 < \sigma_l^2 \\ b_1 + b_2 \sin(\dfrac{\pi(\overline{\sigma}^2 - \sigma_l^2)}{\sigma_u^2 - \sigma_l^2} - \dfrac{\pi}{2}) & \sigma_l^2 \leq \overline{\sigma}^2 \leq \sigma_u^2 \\ b_1 + b_2 & \overline{\sigma}^2 > \sigma_u^2 \end{cases} \tag{6.23}$$

where $b_1, b_2, \sigma_l^2, \sigma_u^2$ are set to constant values. The second condition again ensures smooth transitions between lower and upper limits of $\overline{\sigma}$. As shown in Fig. 6.15 left, a path close to demonstrations (i.e. low variance) has a comparatively wider tunnel region compared to Fig. 6.15 right that represents an area not demonstrated before.

The incremental learning is enabled when the trajectory gets out of the tunnel of VSDS, which means $\bar{\omega}(\boldsymbol{x}_m) < \bar{\omega}_{th}(\boldsymbol{x}_m)$. The incremental learning under GP framework is simply expanding the training dataset for GPR. However, a matrix

Algorithm 4: Incremental learning in 2D space

input : New demonstrations dataset: $\mathbb{D}_n = \{(\boldsymbol{x}_{d,1}, \dot{\boldsymbol{x}}_{d,1}), ..., (\boldsymbol{x}_{d,N}, \dot{\boldsymbol{x}}_{d,N})\}$,
Existed GP dataset: $\mathbb{D}_{gp} = \{(\boldsymbol{x}_{g,1}, \dot{\boldsymbol{x}}_{g,1}), ..., (\boldsymbol{x}_{g,M}, \dot{\boldsymbol{x}}_{g,M})\}$,
Thresholds: $r_{th}, \Delta_1, \Delta_2$

output : updated GP dataset $\mathbb{D}_{gp}$

for $i \leftarrow 1$ **to** N **do**
  for $j \leftarrow 1$ **to** M **do**
    if $\|\boldsymbol{x}_{d,i} - \boldsymbol{x}_{g,j}\| \leq r_{th}$ **then**
      Remove data point $(\boldsymbol{x}_{g,j}, \dot{\boldsymbol{x}}_{g,j})$ from $\mathbb{D}_{gp}$;
      $M = \texttt{Length}\ (\mathbb{D}_{gp})$;
    end
  end
end
for $i \leftarrow 1$ **to** N **do**
  Prediction from GPR:
  $\dot{\boldsymbol{x}}^*_{d,i} = \texttt{GPR}\ (\boldsymbol{x}_{d,i})$;
  if $|\|\dot{\boldsymbol{x}}_{d,i}\| - \|\dot{\boldsymbol{x}}^*_{d,i}\|| \geq \Delta_1$ *or*
  $\texttt{arccos}\ (\frac{\dot{\boldsymbol{x}}_{d,i}\dot{\boldsymbol{x}}^*_{d,i}}{\|\dot{\boldsymbol{x}}_{d,i}\|\|\dot{\boldsymbol{x}}^*_{d,i}\|}) \geq \Delta_2$ **then**
    Add data point $(\boldsymbol{x}_{d,i}, \dot{\boldsymbol{x}}_{d,i})$
    into $\mathbb{D}_{gp}$;
  end
end

inverse computation is done in GPR every time when a new data point is added, which can be computationally inefficient. To deal with this issue, we adapt the trajectory-based sparsity criteria [250] to our context. In particular, we check 1) if new data points should be added in the GP dataset, and 2) if some old data points need to be discarded. This comes from the intuition that each data point in GP is responsible for a certain region around it, named as knowledge region. This region can be imagined as a circle centered at that point in the two-dimensional case. If the new data point is within the knowledge region of the old data point, it implies the old knowledge needs to be updated. The details of the incremental learning are shown in Algorithm 4.

6.3.2 Evaluation

We evaluate our shared control approach in a teleoperation scenario, where we use an Omega.3 haptic device from Force Dimension© as a master interface to control a 7-DOF KUKA robot in Gazebo, that serves as our remote manipulator (Fig. 6.16). The task is that the human teleoperates the KUKA to reach a target object inside the box. First, we show normal task execution, then we demonstrate several scenarios where task knowledge needs to be updated or refined through incremental learning.

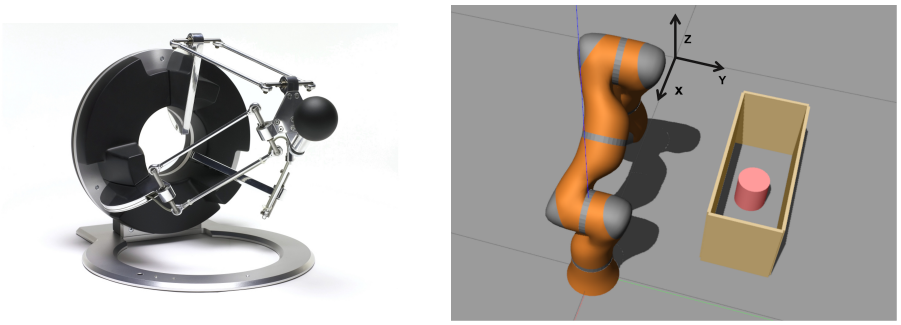


Fig. 6.16: Experiment settings. Left: The 3 DOF omega.3 haptic device used as master interface. Right: The task scenario in Gazebo, with the KUKA LWR as the remote robot and the top surface of the pink object inside the box is the target to reach

Finally, we conduct a user study to compare the performance of our VSDS to other haptic guidance controllers used in previous works, namely an impedance controller tracking a time-indexed trajectory and a flow controller. For simplicity, we constrain the robot motion in x-direction and all the considered motions are in $y-z$ plane.

Normal execution

In this section, we test the ability of our VSDS controller to generate haptic guidance. A human is asked to reach the target object with the robot end-effector via teloperation, while being guided through the force cues. To provide the motion plan, we use the linear DS $\dot{\boldsymbol{x}}_{d,o} = -0.4(\boldsymbol{x}_r - \boldsymbol{x}^*)$, and then locally modulate it with an initial demonstration, with $\gamma_f = 1$ and $l = 0.001$ for the kernel function expressed in (6.20), and $\sigma_n^2 = 0.01$ for the Gaussian noise. The streamlines of the used $\boldsymbol{f}_g$ are shown in Fig. 6.13, left. As for VSDS construction (Fig. 6.13, right), the local attractors are sampled equidistantly from the reference path generated by $\boldsymbol{f}_g$. The stiffness setting is chosen to ensure stable motions on the omega.3 haptic device, where we set $k_{i,1} = 250\text{N/m}$, $a_1 = 1100\text{N/m}$, $a_2 = 700\text{N/m}$, $\sigma_l^2 = 0$, $\sigma_u^2 = 0.85$. As Fig. 6.17 right shows, the human operator is guided to follow the reference path, completing the task without hitting the wall of the box. Fig. 6.17 left shows another scenario where the starting position is different from the demonstration, however, in this particular case the motion plan output of $\boldsymbol{f}_g$ is feasible, and is followed by the human towards the goal location inside the box.

Incremental Learning

In this section, we test the ability of our framework to deal with situations where it is desired to update the task knowledge, or to adapt it due to possible changes in the environment. In the first scenario, the human attempts the reaching task from an initial position far away from demonstrations, and therefore, the governing dynamics are those of the linear DS. This is problematic since while the dynamics converge to the attractor, the path generated leads to collisions with the walls of the box (Fig.

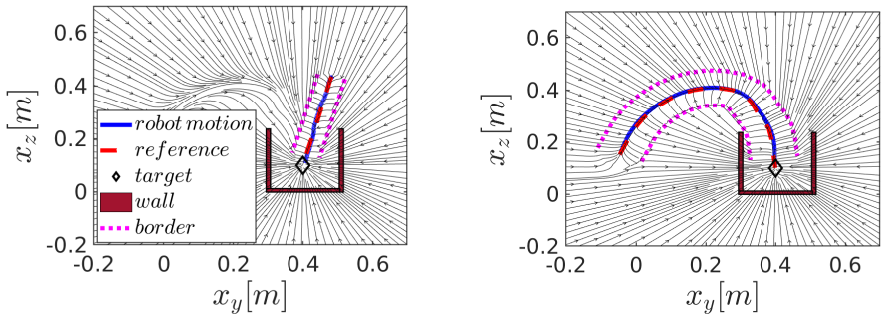
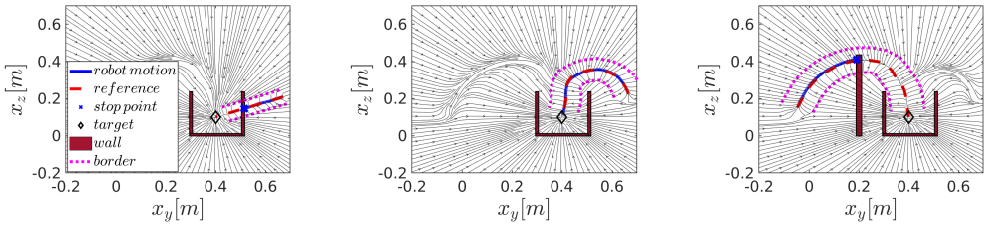
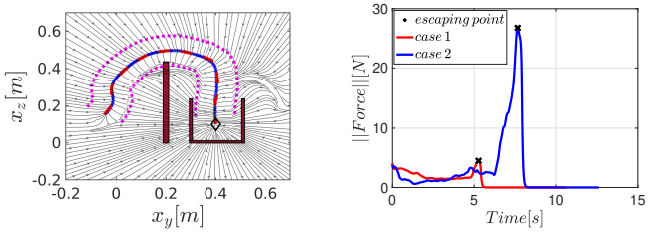


Fig. 6.17: The robot motion for the target-reaching task starting from different initial positions. The blue line is the real robot motion. The red dotted line is the reference motion generated by $\boldsymbol{f}_g$. The pink dotted lines show the border of VSDS tunnel. Left: Starting from a position far away from the demonstration. Right: Starting from a position near the demonstration.



(a) Case 1: before refinement (b) Case 1: after refinement (c) Case 2: before refinement



(d) Case 2: after refinement (e) Escaping Force

Fig. 6.18: Results of motion refinement in two different scenarios, where in one the starting point is far away from the demonstrations (Case 1, Fig. 6.18(a) to Fig. 6.18(b)), while in the other the environment is changed by adding an obstacle (Case 2, Fig. 6.18(c) to Fig. 6.18(d)). For Fig. 6.18(a) to Fig. 6.18(d), the blue path shows the robot motion, the red path is the reference path generated by $\boldsymbol{f}_g$ while the pink dotted lines show the borders of the VSDS tunnel. Fig. 6.18(e) shows the force to escape from VSDS tunnel, where the red plot corresponds to Case 1, while the blue is for Case 2

6.18(a)). As soon as the task execution starts, the human quickly realizes that the guidance is leading him/her in a wrong manner, and therefore exerts a force to escape from the tunnel region of local attraction, where (s)he can then freely manipulate the master device to demonstrate the successful task execution. After the refinement,

when the human starts from the same initial position, (s)he is guided correctly to achieve the task (Fig. 6.18(b)).

We showcase the second scenario in a situation where an obstacle is introduced in a region demonstrated before, and therefore model knowledge should be adapted. As can be seen from Fig. 6.18(c), the streamlines lead to collision with the placed obstacle. The human realizes that (s)he is being guided in the wrong manner, escapes the tunnel region of the guidance and adds a new demonstration to how the collision with the obstacle should be avoided. After the refinement, the human is properly guided along a path that avoids the obstacle (Fig. 6.18(d)).

It should be noted that due to the variable stiffness and the tunnel settings, the required force to escape from the VSDS tunnel differs depending on the region of the state space. In the first case, the human attempts to update task knowledge in a region far away from demonstrations. Therefore, the stiffness is lower and the tunnel region is narrower, and in consequence the force needed to escape the guidance is much lower, compared to the second case, where the obstacle is placed in an area demonstrated before, resulting in a much higher force necessary to escape the guidance (Fig. 6.18(e)).

Finally, it is worth noting also that we can handle motion refinement in both cases, because of our specific choice of the incremental learning method as described in Algorithm 4. More specifically, we assign a knowledge region for each data point, and discard old data points if their knowledge region is shared with new demonstration points. This implies that existing task knowledge is obsolete and should be refined, which is the case for the obstacle scenario (Case 2 in Fig. 6.18).

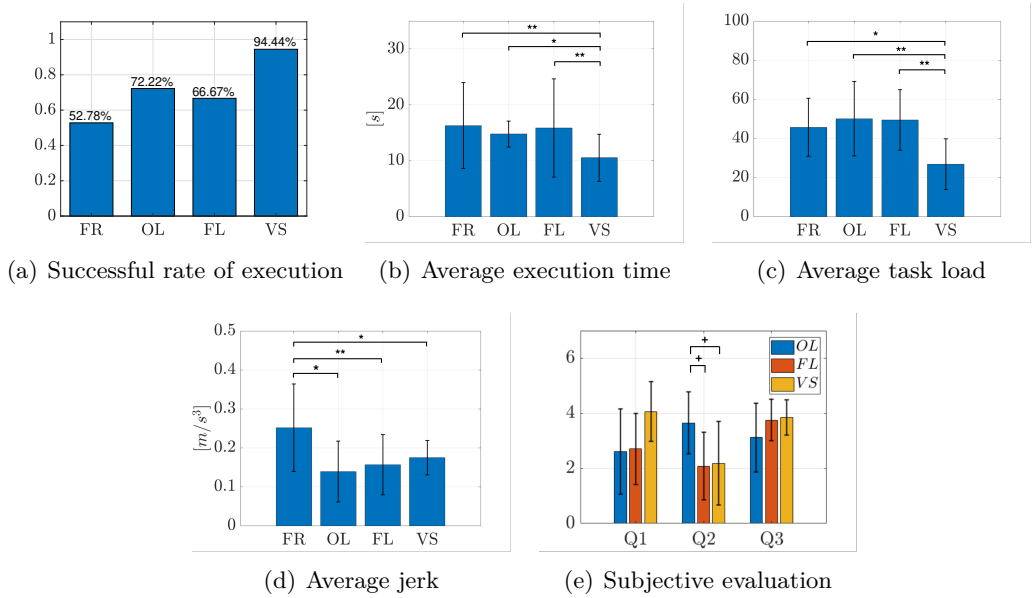
6.3.3 User Study

In this section, we conduct a user study to compare the performance of several controllers for haptic guidance generation, in a target-reaching task.

Methods

The DS shown in Fig. 6.13 left is used to provide the motion plan, where the user starts from an initial position close to the start of the demonstrations to make the remote robot end-effector reach a desired goal location. To provide force cues, we compare the following controllers:

- Our VSDS controller, with streamlines shown in Fig. 6.13, right. (VS mode)
- The Flow controller presented in [131], and described in sec 6.2.1 where $\boldsymbol{u}_c = \boldsymbol{D}_f(\boldsymbol{x}_m)(\boldsymbol{v}_d - \dot{\boldsymbol{x}}_m)$, with streamlines according to Fig. 6.13 left, and where $\boldsymbol{D}_f(\boldsymbol{x}_m)$ is the feedback gain, while $\boldsymbol{v}_d$ is the mapping of $\boldsymbol{f}_g(\boldsymbol{x}_r)$ on the master side. Note that the controller formulation is also similar to the commonly used flow controllers in the exoskeleton literature (e.g. [207]). (FL mode)
- An impedance controller tracking a trajectory $\boldsymbol{x}_d(t)$ integrated in open-loop from $\boldsymbol{f}_g(\boldsymbol{x}_r)$ and mapped to the master, starting from the initial robot position,



(a) Successful rate of execution (b) Average execution time (c) Average task load

(d) Average jerk (e) Subjective evaluation

Fig. 6.19: Results of user study. FR: Free Mode (no guidance), OL: Open-Loop Impedance controller, FL: FLow controller, VS: VSDS controller. Q1, Q2 and Q3 refer to the three questions of the GQ questionnaire. Error bars indicate the standard deviation. To indicate significance between conditions, '**' represents $p < 0.01$, '*' represents $p < 0.05$, '+' represents $p < 0.1$

such that $\boldsymbol{u}_c = \boldsymbol{K}_c(\boldsymbol{x}_d(t) - \boldsymbol{x}_m) - \boldsymbol{D}_c\dot{\boldsymbol{x}}_m$, with $\boldsymbol{K}_c$ and $\boldsymbol{D}_c$ as stiffness and damping. (OL mode)

- Free mode: Teleoperation without guidance. (FR mode)

For the first and third conditions, we use the same constant stiffness matrix. Also, for the VSDS controller, we deactivate the tunnel region effect, since incremental learning is not needed during the user study. This means that the symmetric attraction is active in the entire state space. For the second condition, we set the eigenvalues of $\boldsymbol{D}_f$ to 45 and 20, to provide roughly similar guidance strength to the other two conditions.

We tested 12 participants in total, aged from 20 to 30, with no previous experience in teleoperation. We asked them to interact with the master device to teleoperate the remote robot end-effector to guide it to the pink object inside the box as shown in Fig. 6.16. Subjects could visually observe the motion of the KUKA LWR in Gazebo during teleoperation in real-time. Subjects are instructed to focus primarily on attempting task execution without any collisions, and if possible to be quick, while roughly following a continuous curve towards the goal. Before starting the experiment, we show each participant how to do the task, and give them a familiarisation trial under each condition. During the experiment, participants are asked to conduct three trials for each condition, the sequence of which is randomly shuffled across subjects. After each condition, subjects are requested to fill in NASA TLX and the GQ questionnaire

on Guidance Quality [203] (used also in sec. 5.4.2). After finishing all the trials, we asked participants which condition they preferred the most. To further evaluate the performance, we additionally compute the metrics: successful rate of execution, execution time, the task load computed by using NASA TLX scores, and the jerk of the remote robot movement. We define a trial as successful if the robot reaches the target without hitting the box or the ground, otherwise it is defined as failure.

Data Analysis

With respect to the successful rate, we count the total number of successful trials as a percentage of the total number of trials for each controller. For the remaining metrics, we computed the mean across trials for further statistical analysis [259]. We first tested the data for normality using the Shapiro-Wilk test. Then, we computed repeated measures ANOVAs for normally distributed data, and Friedman test otherwise. We also used Friedman test to analyze the results of the GQ questionnaire since the data is not continuous. This was followed by Bonferroni corrected post-hoc pairwise comparisons to compare the individual conditions. A Greenhouse-Geisser correction was used when the assumption of sphercity was violated, where we used the Mauchly test for sphercity. For the GQ questionnaire, although we recorded the subject response for all conditions, we thought it would be meaningful to analyze the results for the conditions where the guidance is activated, therefore excluding the Free mode. We set the Alpha level to 0.05, where $p < 0.05$ is considered statistically significant, while $p < 0.1$ indicates a statistical tendency.

Results

The results of the user study are shown in Fig. 6.19 as bar plots showing the mean across conditions and the standard deviation, as well as the statistically significant different conditions. Friedman test revealed that all the three guidance conditions reduced the jerk compared to the free mode ($\chi^2 = 13.8, p = 0.003$) with no significant difference across conditions (Fig. 6.19(d)). Friedman test for the execution time also showed significant effects ($\chi^2 = 15.7, p = 0.0013064$), where the VS condition was found to reduce the execution time compared to the FR condition ($p = 0.003$), the OL ($p = 0.04$) and the FL ($p = 0.003$) (Fig. 6.19(b)). For the TLX load, Repeated Measures Anova also showed significant effects ($F(3, 44) = 5.8323, p = 0.0019111$), which mainly were due to the VS condition reducing the task load compared to the other conditions (Fig. 6.19(c)). The evaluation of GQ questionnaire is shown in (Fig. 6.19(e)). The response from the first question ($\chi^2 = 4.7692, p = 0.092$) regarding guidance usefulness indicates VS condition has no significant difference in comparison with others, with $p = 0.13$ compared to OL, and and $p = 0.11$ compared to FL. For Q2, we had ($\chi^2 = 7.0556, p = 0.02937$) mainly caused by a tendency for the OL to have higher scores compared to the VS($p = 0.075$) and the FL ($p = 0.0553$) conditions. On the other hand, no significant differences for Q3 regarding the degree to which subjects felt in control among conditions was found. Finally, the answers of the participants regarding their guidance preference were as follows: VSDS controller

(75%), flow controller (17%), and free mode (8%).

The results of the user study came in line with previous shared control literature that haptic guidance improves the teleoperation performance [203, 209], revealed mainly by higher success rates and lower jerk. The VSDS controller shows the highest rate in comparison with the other two controllers. The relatively higher failure rate for the open-loop impedance controller could be due to the fact that this controller lacks the timing freedom, and therefore, if the user does not attempt to synchronize with the guidance or passively follow it, the results might be unpredictable. On the other hand, the flow controller does not attempt to pull the user to a specific path that successfully achieves the task, but rather follows the streamlines of $\boldsymbol{f}_g$ to reach the target, and therefore following a streamline that collides with the outside of the box is more likely.

We also think that these are the reasons why the open-loop and flow controllers had higher NASA TLX load scores compared to VSDS. The lack of timing freedom in the open-loop impedance controller meant that the subject had to spend additional effort to actively synchronize or even fight against the guidance at times. This is also reflected by noting that the open-loop controller resulted in the highest score in the answer to Q2 (Fig. 6.19(e)), related to fighting the guidance. On the other hand, the higher score for NASA TLX recorded for the flow controller could be due to the fact that subjects did not feel enough restriction to move along a particular path, thereby needed to focus more on moving the end-effector along a collision-free path. The NASA TLX results seem also to be in correlation with the results of Q1 on the usefulness of guidance, with a tendency noticed for the VS condition to have higher scores.

While the VSDS controller generally seemed to have a better performance, in our view, the choice of one haptic guidance approach or another should depend on the given scenario. The OL and VSDS controllers rely essentially on a spring action to provide guidance storing potential energy for large errors from the reference path, thereby makes it more restrictive for the subjects. This would be suitable for example for novice surgeons during training who might lack experience in teleoperation. The flow controller is more forgiving in this regard since the guidance rather provides assistance to move forward along the direction of the flow, but requires more mental demand from the operator to focus on following a collision free path, and in consequence could be useful for more experienced subjects.

Finally, regarding the passivity of our closed loop system, it should be noted since we consider unilateral teleoperation, the only source of potential activity in the system could be due to the haptic guidance controller $\boldsymbol{f}_{vs}(\boldsymbol{x}_m)$, and therefore, ensuring the passivity of the controller would be sufficient to guarantee an overall stable operation. In the next section, we will explore the use of energy tanks from [260] to ensure the passivity and the asymptotic stability of VSDS controllers.

6.4 Passivity-Based VSDS

Unfortunately, the current VSDS formulation suffers from two main drawbacks. First, the robot only practically converges to the global attractor or very close to it, which is also partially dependent on parameter tuning. In other words, there is no theoretical guarantee that the robot is asymptotically stable with respect to the global attractor, which is one of the main features of first-order DS. The second problem is that the velocity profile of the robot can arbitrary differ from the velocity field represented by the motion plan DS $\boldsymbol{f}_g(\boldsymbol{x})$. Ideally, in an open-loop trajectory tracking problem, the velocity of the robot should be similar to that of the desired time-indexed trajectory $\boldsymbol{x}_d(t)$, independently from the values of the stiffness and damping[3]. These impedance parameters should on the other hand mainly affect the robot behavior in physical interaction, in the sense of how it reacts to perturbations or allows deviation from the desired trajectory.
In this section, we will show our proposed approach to solve the two aforementioned problems, namely, tracking the velocity profile of $\boldsymbol{f}_g(\boldsymbol{x})$, and ensuring the asymptotic stability/passivity of the closed loop system. Without loss of generality, in this section, we will assume in the following that $\boldsymbol{x}^*$ is shifted to the origin, such that the desired equilibrium of the system is at $(\boldsymbol{x} = \boldsymbol{0}, \dot{\boldsymbol{x}} = \boldsymbol{0})$.
In the following, the notation p^i indicates the i-th element of the vector $\boldsymbol{p}$, while for a matrix $\boldsymbol{P}^{i,j}$ indicates the element at i-th row and j-th column. We use $\boldsymbol{P}^{i,*}$ for i-th row and $\boldsymbol{P}^{*,j}$ for the j-th column, and $\{\boldsymbol{p}_k\}_{k=1}^K$ to indicate vector elements stacked together evaluated over $k = 1...K$.

6.4.1 Velocity Tracking VSDS

We start by proposing the following new formulation for our VSDS controller

$$\boldsymbol{f}_{vs}(\boldsymbol{x}) = \kappa(||\boldsymbol{x}||)\left(\boldsymbol{f}_{vs,o}(\boldsymbol{x}) + \boldsymbol{f}_f(\boldsymbol{x})\right) + \boldsymbol{\Phi}(\boldsymbol{x}) \tag{6.24}$$

$$\boldsymbol{u}_c = \boldsymbol{f}_{vs}(\boldsymbol{x}) - \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} \tag{6.25}$$

where $\boldsymbol{f}_{vs,o}(\boldsymbol{x}) = \sum_{i=1}^{N} \tilde{\omega}_i(\boldsymbol{x})\, \boldsymbol{f}_{l,i}(\boldsymbol{x})$ is the original VSDS formulation, $\kappa(||\boldsymbol{x}||) = 1 - \mathrm{e}^{-a||\boldsymbol{x}||}$ with $a > 0$ is a smooth activation function that outputs a value of 0 at the equilibrium, $\boldsymbol{\Phi}(\boldsymbol{x})$ is a conservative force field that also vanishes at the equilibrium, while $\boldsymbol{f}_f(\boldsymbol{x})$ is another force field that ensures velocity tracking. The role and design of these controller elements will be further elaborated in the following. As for the damping matrix $\boldsymbol{D}(\boldsymbol{x})$, in this section, we adopt a more elaborate design in order to assign a specific damping relationship to each linear DS i.e

$$\boldsymbol{D}(\boldsymbol{x}) = \sum_{i=1}^{N} \widetilde{\omega}_i(\boldsymbol{x})\boldsymbol{D}_i, \tag{6.26}$$

where $\boldsymbol{D}_i$ is a positive definite matrix.
As stated earlier, in a trajectory-tracking problem, it is desired that the robot not

[3]Under the assumption of a stiffness high enough to overcome robot friction and properly tuned damping.

only follows the geometric path described by the trajectory, but also follows its timing law, which is mainly highlighted by the velocity of the robot being close or identical to that of the desired trajectory. This should also happen independently of the chosen stiffness profile. To achieve this objective and follow the velocity of the trajectory described by $\boldsymbol{f}_g(\boldsymbol{x})$, we design the force field $\boldsymbol{f}_f(\boldsymbol{x})$ from eqn. (6.24) accordingly. This force field can be viewed as a feed forward term, similar to those typically used in computed torque control methods [261], that rely on the desired trajectory higher-order derivatives to ensure trajectory tracking (see sec. 2.2.4). In the following, we propose two different solutions for the design of $\boldsymbol{f}_f(\boldsymbol{x})$ to achieve this desiderata.

Velocity Feedback Approach

The first possibility we explored in this regard was to simply augment our formulation with a velocity tracking term. The control law can be designed with the feed-forward term as:

$$\boldsymbol{f}_f(\boldsymbol{x}) = \boldsymbol{D}(\boldsymbol{x})\boldsymbol{f}_g(\boldsymbol{x}) \tag{6.27}$$

with $\boldsymbol{D}(\boldsymbol{x})$ designed according to (6.26), and where, together with (6.25), we are closing the loop around the velocity tracking error $\dot{\boldsymbol{e}} = \boldsymbol{f}_g(\boldsymbol{x}) - \dot{\boldsymbol{x}}$.
While this solution is simple and straightforward to implement, we noticed that whereas high damping gains lead to better velocity tracking, they result in slightly altering the symmetric spring-like attraction behavior compared to the original VSDS. Furthermore, as it will be shown in the experimental validation, the formulation also results in higher contact forces during external interactions.

Optimization-based design

To alleviate this problem, the second solution we propose is to optimize the feed-forward force fields based on some optimal reference behavior. More specifically, our aim is that a robot driven by a VSDS control law is to follow a reference path, as well as a desired velocity, in a manner similar to time-indexed open-loop trajectory tracking. In other words, in free motion, the simplest form of the optimal target behavior is equivalent to:

$$\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} = \boldsymbol{K}_{des}(\boldsymbol{x})(\boldsymbol{x}_d(t) - \boldsymbol{x}) - \boldsymbol{D}_{des}(\boldsymbol{x})\dot{\boldsymbol{x}}, \tag{6.28}$$

where $\boldsymbol{D}_{des}(\boldsymbol{x})$ is a damping profile computed from the stiffness profile in order to maintain a critically damped system (or to achieve another control objective), while $\boldsymbol{x}_d(t)$ is a time-indexed trajectory generated by the open-loop integration of $\boldsymbol{f}_g(\boldsymbol{x}_d)$ starting from the robot initial position. In principle, we could have included also $\dot{\boldsymbol{x}}_d(t)$ in (6.28); however, we found that (6.28) in its current form results in good tracking results.
We aim to design $\boldsymbol{f}_f(\boldsymbol{x})$ based on (6.28). Assuming that $\boldsymbol{f}_g(\boldsymbol{x})$ and $\boldsymbol{K}_{des}(\boldsymbol{x})$ are available, the second-order system (6.28) can be simulated offline. This results in a data set that consists of $\{\boldsymbol{x}_t\}_{t=0}^{T}$ and $\{\boldsymbol{F}_{s,t}\}_{t=0}^{T}$, with t as the time index and T

is the total simulation time. The simulated position response of the second-order system is $\boldsymbol{x}_t$, while $\boldsymbol{F}_{s,t} = \boldsymbol{K}_{des}(\boldsymbol{x}_t)(\boldsymbol{x}_{d,t} - \boldsymbol{x}_t)$ is the resulting spring force. We can then optimize $\boldsymbol{f}_f(\boldsymbol{x})$ offline based on the collected data. We propose the following structure for $\boldsymbol{f}_f(\boldsymbol{x})$:

$$\boldsymbol{f}_f(\boldsymbol{x}) = \sum_{i=1}^{N} \widetilde{\omega}_i(\boldsymbol{x})\boldsymbol{\Gamma}_i. \tag{6.29}$$

which represents a weighted sum of constant forces $\boldsymbol{\Gamma}_i$. The goal of the optimization is then to tune $\boldsymbol{\Gamma}_i$ to ensure the best possible tracking performance, which can be formulated as

$$\min_{\boldsymbol{\Gamma}_{i,\, i=1,\dots,N}} \quad ||\boldsymbol{f}_{vs}(\boldsymbol{x}_t) - \boldsymbol{F}_{s,t}||^2 \tag{6.30a}$$

$$\text{subject to } \underline{\boldsymbol{\Gamma}}_i \leq \boldsymbol{\Gamma}_i \leq \overline{\boldsymbol{\Gamma}}_i, \tag{6.30b}$$

$$|\boldsymbol{f}_{vs}(\boldsymbol{x}_0)| > \underline{\boldsymbol{F}}, \tag{6.30c}$$

which minimizes the norm between the VSDS term $\boldsymbol{f}_{vs}(\boldsymbol{x}_t)$ over the simulated position response and the resulting spring term $\boldsymbol{F}_{s,t}$ of the second order system. The constraint (6.30b) was added to ensure reasonable upper and lower bounds on the constant force terms. On the other hand, the constraint (6.30c) ensures a high enough initial spring force that can overcome robot friction, which becomes crucial for the implementation of the control policy on the real robot.
To solve (6.30), tools such as `fmincon` provided by Matlab® can be used. This however resulted in a high computation time (around 1 minute) in order to compute the optimal solution, which also increases as the number of linear DS N in the VSDS increases. To improve efficiency, in the following we show that it is possible to formulate our optimization as a convex Quadratic Program (QP). First, let us write the problem (6.30) as

$$\min_{\boldsymbol{\Gamma}_{i,\, i=1,\dots,N}} \quad \| \sum_{i=1}^{N} \kappa(\|\boldsymbol{x}_t\|)\widetilde{\omega}_i(\boldsymbol{x}_t)\boldsymbol{\Gamma}_i + \boldsymbol{f}_s(x_t) - \boldsymbol{F}_{s,t}\|^2 \tag{6.31a}$$

$$\text{subject to } \underline{\boldsymbol{\Gamma}}_i \leq \boldsymbol{\Gamma}_i \leq \overline{\boldsymbol{\Gamma}}_i, \tag{6.31b}$$

$$| \sum_{i=1}^{N} \kappa(||\boldsymbol{x}_0||)\widetilde{\omega}_i(\boldsymbol{x}_0)\boldsymbol{\Gamma}_i + \boldsymbol{f}_s(\boldsymbol{x}_0)| > \underline{\boldsymbol{F}}, \tag{6.31c}$$

where $\boldsymbol{f}_s(\boldsymbol{x}) = \boldsymbol{f}_{vs,o}(\boldsymbol{x}) + \boldsymbol{\Phi}(\boldsymbol{x})$. Defining :

$$\boldsymbol{W}(\boldsymbol{x}) = \begin{bmatrix} \kappa(||\boldsymbol{x}_0||)\widetilde{\omega}_1(\boldsymbol{x}_0) & \dots & \kappa(||\boldsymbol{x}_0||)\widetilde{\omega}_N(\boldsymbol{x}_0) \\ \vdots & \ddots & \vdots \\ \kappa(||\boldsymbol{x}_T||)\widetilde{\omega}_1(\boldsymbol{x}_T) & \dots & \kappa(||\boldsymbol{x}_T||)\widetilde{\omega}_N(\boldsymbol{x}_T), \end{bmatrix} \tag{6.32a}$$

$$\boldsymbol{W}_a = \begin{bmatrix} \boldsymbol{W}(\boldsymbol{x}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}(\boldsymbol{x}) \end{bmatrix}, \boldsymbol{W}_{a,0} = \begin{bmatrix} \sigma_1 \boldsymbol{W}^{1,*} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_2 \boldsymbol{W}^{1,*} \end{bmatrix}, \tag{6.32b}$$

$$\underline{\boldsymbol{\Gamma}}_{sh} = \begin{bmatrix} \underline{F}^1 - \sigma_1 f_s^1(\boldsymbol{x}_0) \\ \underline{F}^2 - \sigma_2 f_s^2(\boldsymbol{x}_0) \end{bmatrix} \tag{6.32c}$$

$$\boldsymbol{F}_{sh,t} = \boldsymbol{F}_{s,t} - \boldsymbol{f}_s(\boldsymbol{x}_t) \tag{6.32d}$$

$$\boldsymbol{F}_{sh} = \begin{bmatrix} \{F_{sh,t}^1\}_{t=1}^T & \{F_{sh,t}^2\}_{t=1}^T \end{bmatrix} \tag{6.32e}$$

Generally, we have that $\boldsymbol{W} \in \mathbb{R}^{T \times N}$, $\boldsymbol{W}_a \in \mathbb{R}^{mT \times mN}$, $\boldsymbol{W}_{a,0} \in \mathbb{R}^{m \times mN}$, $\underline{\boldsymbol{\Gamma}}_{sh} \in \mathbb{R}^m$ and $\boldsymbol{F}_{sh} \in \mathbb{R}^{mT}$. For ease of illustration, we chose to formulate all the above terms with $m = 2$. The binary variable σ_i (where $i = 1, 2$) is defined as

$$\sigma_i = \begin{cases} 1 & f_s^i(\boldsymbol{x}_0) \geq 0 \\ -1 & f_s^i(\boldsymbol{x}_0) < 0 \end{cases} . \tag{6.33}$$

we can then rewrite (6.31) as

$$\min_{\boldsymbol{y}} \quad (\boldsymbol{W}_a \boldsymbol{y} - \boldsymbol{F}_{sh,t})^T (\boldsymbol{W}_a \boldsymbol{y} - \boldsymbol{F}_{sh,t}) \tag{6.34a}$$

$$\text{subject to } \underline{\boldsymbol{y}} \leq \boldsymbol{y} \leq \overline{\boldsymbol{y}}, \tag{6.34b}$$

$$\boldsymbol{W}_{a,0} \boldsymbol{y} > \underline{\boldsymbol{\Gamma}}_{sh}, \tag{6.34c}$$

with $\boldsymbol{y} = \begin{bmatrix} \{\Gamma^1\}_{i=1}^N & \{\Gamma^2\}_{i=1}^N \end{bmatrix}$, $\underline{\boldsymbol{y}} = \begin{bmatrix} \{\underline{\Gamma}_i^1\}_{i=1}^N & \{\underline{\Gamma}_i^2\}_{i=1}^N \end{bmatrix}$ and $\overline{\boldsymbol{y}} = \begin{bmatrix} \{\overline{\Gamma}_i^1\}_{i=1}^N & \{\overline{\Gamma}_i^2\}_{i=1}^N \end{bmatrix}$ containing the elements of all m DOF concatenated. The introduction of the binary variable (6.33) is necessary to realize the constraint (6.31c) which can be reformulated as $\sum_{i=1}^N \kappa(||\boldsymbol{x}_0||)\widetilde{\omega}_i(\boldsymbol{x}_0)\boldsymbol{\Gamma}_i > \underline{\boldsymbol{F}} - \boldsymbol{f}_s(\boldsymbol{x}_0)$ if $\boldsymbol{f}_s(\boldsymbol{x}_0) \geq 0$, and $\sum_{i=1}^N -\kappa(||\boldsymbol{x}_0||)\widetilde{\omega}_i(\boldsymbol{x}_0)\boldsymbol{\Gamma}_i > \underline{\boldsymbol{F}} + \boldsymbol{f}_s(\boldsymbol{x}_0)$ otherwise. Since both constraints cannot be active at the same time, the role of σ_i is to activate one of these constraints depending on the sign of the initial force $\boldsymbol{f}_s(\boldsymbol{x}_0)$, resulting in the constraint formulated in (6.34c).
Expanding further, and through some simple manipulations, we can reformulate (6.34) as

$$\min_{\boldsymbol{y}} \quad \frac{1}{2} \boldsymbol{y}^T \boldsymbol{H} \boldsymbol{y} + \boldsymbol{c}^T \boldsymbol{y} \tag{6.35a}$$

$$\text{subject to } \boldsymbol{A}\boldsymbol{y} \geq \boldsymbol{b}, \tag{6.35b}$$

$$\underline{\boldsymbol{y}} \leq \boldsymbol{y} \leq \overline{\boldsymbol{y}}. \tag{6.35c}$$

where $\boldsymbol{H} = \boldsymbol{W}_a^T \boldsymbol{W}_a$, $\boldsymbol{c} = -\boldsymbol{W}_a^T \boldsymbol{F}_{sh}$, $\boldsymbol{A} = \boldsymbol{W}_{a,0}$, $\boldsymbol{b} = \underline{\boldsymbol{\Gamma}}_{sh}$, which represents the well-known form of a QP program. It is worth mentioning that, when expanding

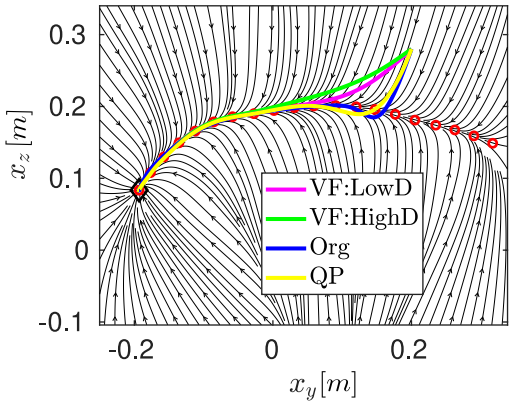


Fig. 6.20: Comparison between the symmetric attraction behavior of the original VSDS (Org), the optimization based approach (QP), the velocity feedback method with low (VF:LowD) and high (VF:HighD) damping gains. ©2023 IEEE [20].

(6.34a), a constant term that does not depend on $\boldsymbol{y}$ has been omitted, since it does not affect the optimization. Finally, it is worth noting that the reformulation of the optimization as a QP program results in a substantial reduction of the computation time for the optimization, which now gets solved in approximately 0.1 s, for $N = 20$. Figure 6.20 shows the resulting attraction behavior of a simulated mass slightly perturbed from the reference path, and driven by a control law designed based on *i)* the original VSDS formulation $\boldsymbol{f}_{vs,o}(\boldsymbol{x})$, *ii)* the QP-based optimization (eq. (6.29)) and Velocity feedback (eq. (6.27)) methods for the design of $\boldsymbol{f}_f(\boldsymbol{x})$, with the eigenvalues of $\boldsymbol{D}(\boldsymbol{x})$ where set once to *iii)* low and once to *iv)* high. While the behavior with the original approach and the QP methods are similar, for the velocity feedback method, increasing the damping gain delays the contact point between the mass and the reference path. This can be attributed to $\boldsymbol{f}_g(\boldsymbol{x})$ which naturally points towards the global attractor. Increasing $\boldsymbol{D}(\boldsymbol{x})$ magnifies this behavior and neutralizes the effect of $\boldsymbol{f}_{vs,o}(\boldsymbol{x})$ which aims to pull the mass back to the reference path in a spring-like manner, and therefore interferes to some extent with the original VSDS dynamics.

6.4.2 Energy Tank-based Control

The VSDS formulation developed in Sec. 6.4.1 is exploited to drive the robot in closed-loop, i.e., we need to explicitly take into account the robot dynamics to derive stability results. In general, the closed-loop implementation of VSDS is not guaranteed to drive the robot towards a desired equilibrium neither to ensure a stable interaction with a passive environment. In this section, we analyze the stability and passivity properties of VSDS and exploit the energy tanks formalism to ensure *i)* asymptotic stability in free motion and *ii)* a passive interaction behavior.

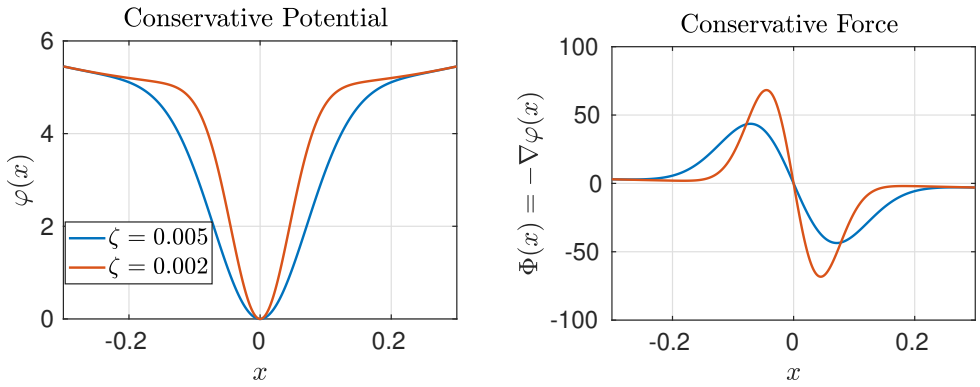


Fig. 6.21: Left: Illustration of the conservative potential over 1 DOF for two values of ζ. Right: The corresponding force field derived as the gradient of the potential. In both cases, we use $\kappa_o = 5$ and $\tau_{min} = 5$. ©2023 IEEE [20].

Passivity analysis

We analyze the passivity of the system under the VSDS control law (6.24). The closed-loop dynamics obtained by substituting (6.25) and (6.24) into the robot dynamics (6.1) is

$$\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} = -\boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{u}_e + \boldsymbol{f}_{vs} - \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}}, \tag{6.36}$$

where $\boldsymbol{D}(\boldsymbol{x})$ is positive definite if each $\boldsymbol{D}_i$ is a positive definite matrix.
We consider the storage function

$$\mathcal{W}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \frac{1}{2}\dot{\boldsymbol{x}}^T\boldsymbol{M}(\boldsymbol{x})\dot{\boldsymbol{x}} + \varphi(\boldsymbol{x}), \tag{6.37}$$

where the first term on the right side is the kinetic energy and $\varphi(\boldsymbol{x})$ is the potential function that generates the conservative field $\boldsymbol{\Phi}(\boldsymbol{x})$, i.e., $\boldsymbol{\Phi}(\boldsymbol{x}) = -\nabla\varphi(\boldsymbol{x})$, and where we have $\varphi(0) = 0$ and $\varphi(\boldsymbol{x}) > 0\ \forall \boldsymbol{x} \neq 0$. We design $\varphi(\boldsymbol{x})$ as

$$\varphi(\boldsymbol{x}) = k_o(1 - e^{-\frac{\boldsymbol{x}^T\boldsymbol{x}}{2\zeta}}) + \tau_{min}\boldsymbol{x}^T\boldsymbol{x} \tag{6.38}$$

where τ_{min}, k_o and ζ are positive constants. In principle, we could have used a simple potential with a constant spring $\boldsymbol{K}_c$ such that $\varphi = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{K}_c\boldsymbol{x}$. This design however would interfere with the VSDS dynamics and the desired stiffness behavior specified by $\boldsymbol{K}_{des}$. On the other hand, the choice in (6.38) allows us to selectively tune the effect of the conservative potential. For example, we can choose to have a weak influence for the potential in regions far away from the equilibrium by choosing a low τ_{min}, while smoothly transitioning to have stronger influence in a small neighborhood around the attractor to ensure convergence. The width and strength of this neighborhood are controlled by the parameters ζ and k_o, as depicted in Fig. 6.21.

Taking the time derivative of $\mathcal{W}$ and considering the expression of $\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}}$ in (6.36)

we obtain

$$\begin{aligned}\dot{\mathcal{W}} &= \dot{\boldsymbol{x}}^T \boldsymbol{M}\ddot{\boldsymbol{x}} + \nabla\varphi^T\dot{\boldsymbol{x}} \\ &= -\dot{\boldsymbol{x}}^T \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} + \dot{\boldsymbol{x}}^T \boldsymbol{u}_e + \kappa\dot{\boldsymbol{x}}^T \left(\boldsymbol{f}_{vs,o} + \boldsymbol{f}_f\right),\end{aligned} \tag{6.39}$$

where we used the property that $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ is skew-symmetric. The sign of $\kappa\dot{\boldsymbol{x}}^T \left(\boldsymbol{f}_{vs,o} + \boldsymbol{f}_f\right)$ is undefined and does not allow to conclude the passivity of the system.

Energy tank-based passification

We resort to the concept of energy tanks [149, 260] to render to closed-loop dynamics (6.36) passive. To this end, we consider an energy storing element with storage function $s(\boldsymbol{x}, \dot{\boldsymbol{x}})$. The dynamics of s is defined as

$$\dot{s} = \alpha(s)\dot{\boldsymbol{x}}^T \boldsymbol{D}(\boldsymbol{x})\dot{\boldsymbol{x}} - \beta(z, s)z - (\eta - \kappa(\|\boldsymbol{x}\|))s, \tag{6.40}$$

where $z = \kappa(\|\boldsymbol{x}\|)\dot{\boldsymbol{x}}^T \left(\boldsymbol{f}_{vs,o} + \boldsymbol{f}_f\right)$ is the term with undefined sign in (6.39). The novelty in (6.40) compared to classical energy tank formulations (e.g. sec. 2.2.2) lies in the term $-(\eta - \kappa(\|\boldsymbol{x}\|))s$ with $\eta > 1$ which ensures that $s \to 0$ for $\dot{\boldsymbol{x}} = 0$. This property is exploited in Sec. 6.4.2 to show the asymptotic stability of the closed-loop system. Assuming that $\eta \approx 1$ also reduces the effects of $-(\eta - \kappa(\|\boldsymbol{x}\|))s$ on tank dynamics far from the position equilibrium ($\boldsymbol{x} = 0$) while ensuring a rapid convergence of s approaching the desired state ($\kappa(0) = 0$). The variables $\alpha(s)$ and $\beta(z, s)$ satisfy

$$\begin{cases} 0 \leq \alpha(s) < 1 & s < \overline{s} \\ \alpha(s) = 0 & \text{otherwise} \end{cases}, \tag{6.41}$$

and

$$\begin{cases} \beta(z, s) = 0 & s \geq \overline{s} \text{ and } z < 0 \\ \beta(z, s) = 0 & s \leq 0 \text{ and } z \geq 0 \\ 0 \leq \beta(z, s) \leq 1 & \text{otherwise} \end{cases}, \tag{6.42}$$

where $\overline{s}$ is the maximum allowed energy. This definition of $\alpha(s)$ and $\beta(z, s)$ ensures that $s \geq 0$ everywhere if the initial energy $s_0 \geq 0$. Therefore, we can add s to the storage function in (6.37) as

$$\mathcal{W}(\boldsymbol{x}, \dot{\boldsymbol{x}}, s) = \frac{1}{2}\dot{\boldsymbol{x}}^T \boldsymbol{M}(\boldsymbol{x})\dot{\boldsymbol{x}} + \varphi(\boldsymbol{x}) + s. \tag{6.43}$$

By construction, the storage function in (6.43) is positive definite, radially unbounded, and vanishes at the equilibrium $(\boldsymbol{x}, \dot{\boldsymbol{x}}, s) = (0, 0, 0)$. To passify the closed loop dynamics, we rewrite the control law (6.24) as

$$\boldsymbol{f}_{vs}(\boldsymbol{x}) = \boldsymbol{\Phi}(\boldsymbol{x}) + \gamma(z, s)\kappa(||\boldsymbol{x}||) \left(\boldsymbol{f}_{vs,o}(\boldsymbol{x}) + \boldsymbol{f}_f(\boldsymbol{x})\right) \tag{6.44}$$

where

$$\begin{cases} \gamma(z, s) = \beta(z, s) & z \geq 0 \\ \gamma(z, s) \geq \beta(z, s) & \text{otherwise} \end{cases}. \tag{6.45}$$

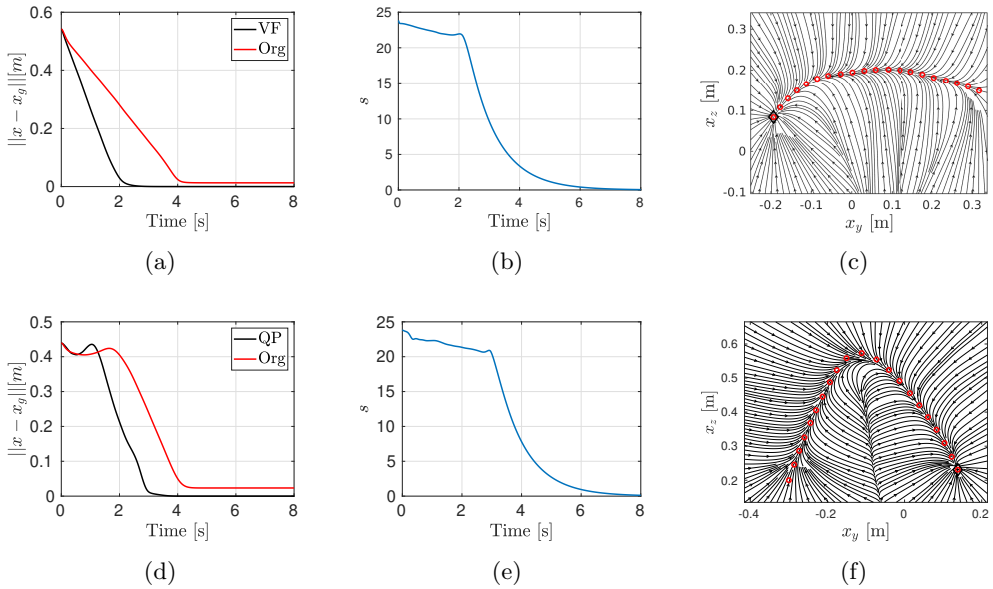


Fig. 6.22: Simulation results for the VSDS controller applied on a simulated mass. The first row shows the comparison results of the original VSDS controller (Org in the legend), to the velocity feedback formulation for computing the feed forward terms (VF), applied on a curve motion. The second row shows the comparison between the orignal formulation and the QP-optimization approach (QP), applied on an angle shaped motion. The streamlines of the VSDS dynamics are shown in the last column, with red dots depicting the local attractors, and the rhombus the global equilbruim. ©2023 IEEE [20].

By taking the time derivative of (6.43), it holds that

$$\begin{aligned}\dot{\mathcal{W}} &= \dot{\boldsymbol{x}}^T \boldsymbol{M} \ddot{\boldsymbol{x}} + \nabla\varphi^T \dot{\boldsymbol{x}} + \dot{s} \\ &= -\dot{\boldsymbol{x}}^T \boldsymbol{D}(\boldsymbol{x}) \dot{\boldsymbol{x}} + \dot{\boldsymbol{x}}^T \boldsymbol{u}_e + \gamma(z, s) z + \dot{s}\end{aligned} \tag{6.46}$$

By substituting (6.40) in (6.46), we obtain

$$\begin{aligned}\dot{\mathcal{W}} =& \dot{\boldsymbol{x}}^T \boldsymbol{u}_e + (\alpha - 1)\dot{\boldsymbol{x}}^T \boldsymbol{D}(\boldsymbol{x}) \dot{\boldsymbol{x}} + (\gamma - \beta) z \\ &- (\eta - \kappa(\|\boldsymbol{x}\|)) s \leq \dot{\boldsymbol{x}}^T \boldsymbol{u}_e,\end{aligned} \tag{6.47}$$

where owing to the definitions of α, γ and β in (6.41), (6.45) and (6.42), respectively, we always have $s \geq 0$, and in consequence the sign of the last three terms in $\dot{\mathcal{W}}$ is negative semi-definite. This results in $\dot{\mathcal{W}} \leq \dot{\boldsymbol{x}}^T \boldsymbol{u}_e$, and we can therefore conclude the passivity of the closed-loop system with respect to the port $(\dot{\boldsymbol{x}}, \boldsymbol{u}_e)$ through which the robot interacts with the external environment.

Stability analysis of the passification control

The storage function (6.43) is radially unbounded, positive definite, and vanishes at the equilibrium. Hence, it can be used as candidate Lyapunov function to verify

the asymptotic stability of the system in the absence of external forces (i.e., $\boldsymbol{u}_e = 0$). It is worth mentioning that the passivity of the system is sufficient to ensure stability. Indeed, $\dot{\mathcal{W}}$ in (6.47) is negative semi-definite as it vanishes for $(\dot{\boldsymbol{x}}, s) = (0, 0)$ irrespective of the value of $\boldsymbol{x}$, i.e., $\boldsymbol{u}_e = 0 \rightarrow \dot{\mathcal{W}} \leq 0, \forall(\boldsymbol{x}, 0, 0)$. Here, we show that the system asymptotically converges to the desired goal (assumed to be $(\boldsymbol{x}, \dot{\boldsymbol{x}}, s) = (0, 0, 0)$ without loss of generality).

Let's assume that $\boldsymbol{u}_e = 0$, $\dot{\boldsymbol{x}} = 0$, and $s = 0$, this also implies that $z = 0$. The closed-loop dynamics (6.36) becomes

$$\ddot{\boldsymbol{x}} = \boldsymbol{M}(\boldsymbol{x})^{-1} \left(\boldsymbol{\Phi}(\boldsymbol{x}) + \gamma\kappa \left(\boldsymbol{f}_{vs,o}(\boldsymbol{x}) + \boldsymbol{f}_f(\boldsymbol{x}) \right) \right), \tag{6.48}$$

From the definition of γ in (6.45), we have that $\gamma = \beta$ if $z = 0$. Moreover, being $s = 0$, we have from (6.42) that $\gamma = \beta = 0$ and that (6.48) vanishes if and only if $\boldsymbol{M}(\boldsymbol{x})^{-1}\boldsymbol{\Phi}(\boldsymbol{x}) = 0 \leftrightarrow \boldsymbol{x} = 0$. The LaSalle's invariance principle (theorem 2) can be used to conclude the asymptotic stability [220].

Simulation results

To verify the validity of the above theoretical concepts, we conduct a series of simulations, where we tested the developed controllers on a simulated mass. In particular, we constructed our VSDS controller with a first-order DS learnt based on two different types of motions. For the angle shaped motion of the LASA HandWriting dataset [262], we compare the performance of the original VSDS approach in (eqn. (6.14)), with the new VSDS formulation (eqn.(6.44)) where the feedforward term was designed via the QP optimization (eqn.(6.29)). On the other hand, we compare the original VSDS with the feedforward terms designed based on the velocity feedback approach (eqn.(6.27)) for the curve shape. As shown in Fig. 6.22, the new formulation results in asymptotic convergence of the mass to the global attractor regardless of the method used for computing the feedforward terms, in comparison to a very low steady state error in the original formulation. Figures 6.22(e) and 6.22(b) show the tank state s, where it could be noted that the tank is slowly depleted in the beginning of the motion, followed by a more rapid convergence when the linear dynamics become more dominant close to the equilibrium.

6.4.3 Results

In this section, we conduct a series of experiments in order to validate our approach in terms of accurate motion execution, safety and physical interaction tasks. The validation is performed on a 7-DOF Kuka LWR, controlled via a Desktop PC with a Core i7. The robot is commanded via the Fast Research interface library, in Cartesian Impedance control mode where the computed control law $\boldsymbol{u}_c$ is sent to the robot as a feed-forward force. Unless otherwise stated, in the following evaluations we used $N = 20$ linear DS. For the potential $\boldsymbol{\Phi}(\boldsymbol{x})$, we set $\zeta = 0.006$, $\tau_{min} = 1$, while k_o needed slight tuning depending on the VSDS approach used, but was typically set in the range of 0.3 to 0.5. Similarly, the damping gains $\boldsymbol{D}_i$ used in (6.27) for the

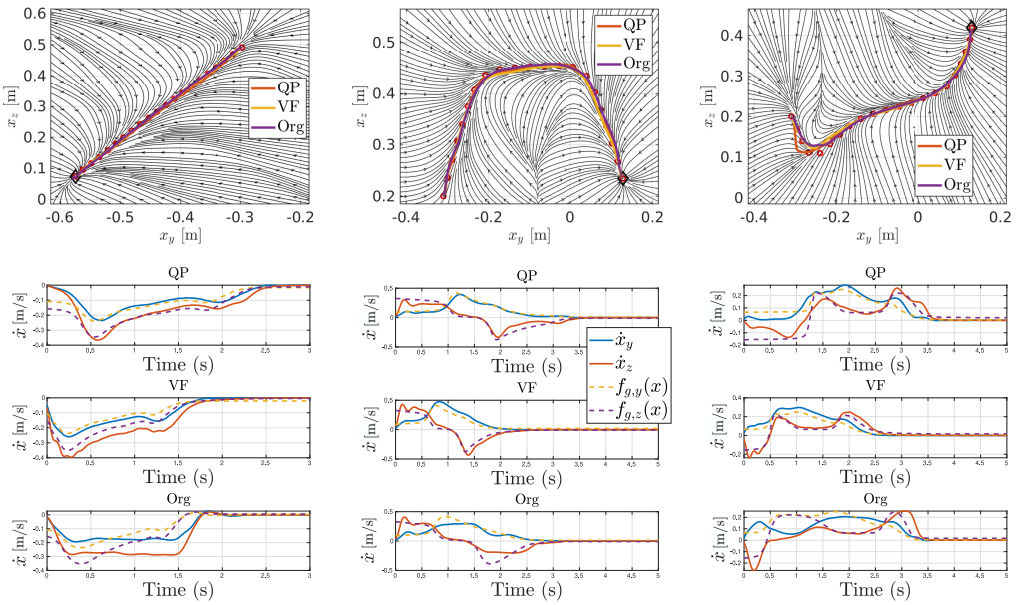
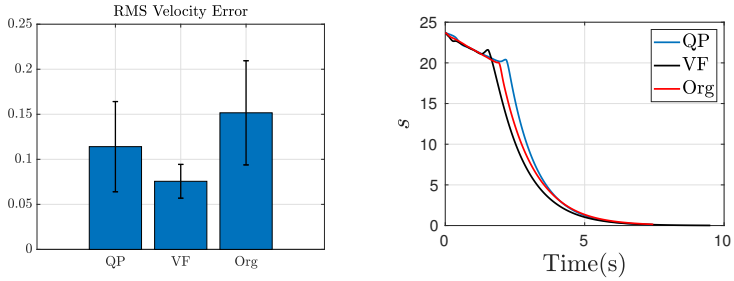


Fig. 6.23: Experimental results for the controllers comparisons for motion execution. The first row shows the spatial position of the executed motions for the constant stiffness case, with the streamlines of the VSDS dynamics in the background. The lower row shows the corresponding velocity profile with solid lines as the actual velocity and dotted as the desired, for the y- and z- directions. ©2023 IEEE [20].

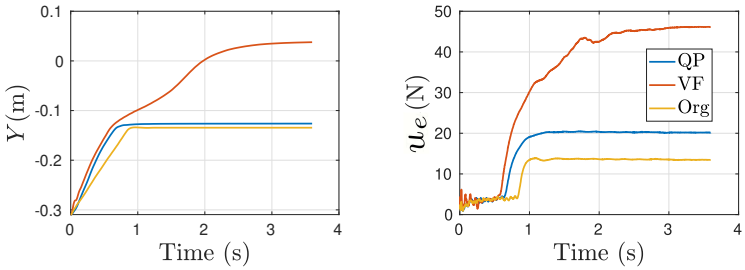


(a) RMS Velocity Error (b) Tank state

Fig. 6.24: Left: Mean RMS velocity error over all executed motions for each controller. Right: comparison of the tank states from the three controllers, for the same executed motion type and stiffness condition. ©2023 IEEE [20].

Velocity Feedback Method were set in the range $200 - 300$ N/m^2, depending on the motion type to be executed. For the tank, we used $s_0 = 30$ and $\eta = 1.05$. Based on practical experience, we set the values of $\underline{\boldsymbol{F}}$ in the optimization problem (6.35) to 10N, which represents a high enough initial force to allow a Kuka LWR to move.

(a) Initial Config. (b) Final Config. Org (c) Final Config. QP (d) Final Config. VF



(e) Position over time (f) External Force

Fig. 6.25: Results from the Collision experiments. Top row shows a snapshot from the setup at the beginning of the experiment, as well as a snapshot from the final configuration for each of the compared approaches. The bottom row shows the position and the norm of the external force for the three approaches. ©2023 IEEE [20].

Motion Execution

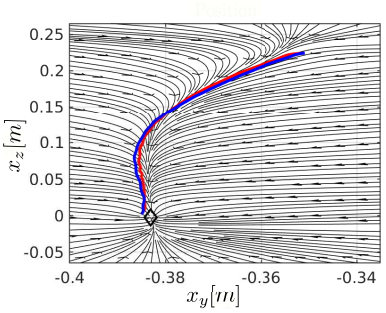
In the first part of the validation, we test the ability of our controller to execute motions following a desired path and a reference velocity profile, while also asymptotically converging to the global attractor. We use eqn.(6.44) to compute the VSDS force component $\boldsymbol{f}_{vs}(\boldsymbol{x})$, subsequently used for commanding $\boldsymbol{u}_c$ as in eqn.(6.25). We compare with $\boldsymbol{f}_f(\boldsymbol{x})$ designed based on Velocity Feedback (VF) (eqn. (6.27)), the optimization-based design (QP) (eqn. (6.29)) and the original VSDS approach (Org) with $\boldsymbol{f}_f(\boldsymbol{x}) = \boldsymbol{0}$.

As stated earlier, the specification of the stiffness should be independent from the velocity profile of the robot. Therefore, we compare the motion execution for two stiffness profiles: a constant stiffness with $\boldsymbol{K}_{des} = diag(1200, 1500)$ and a state-varying stiffness with diagonal elements set to $950 + 150\sin(15\boldsymbol{x}^1 + 0.8)$ and $1200 + 200\sin(15\boldsymbol{x}^1 + 0.8)$. Finally, we test three motions from the LASA Handwriting Data Set with increasing levels of complexity [262]: a straight line, a trapezoidal motion and a Khamesh Shape. The motion data was first appropriately scaled and shifted to make it feasible for robot execution, then we used the approach from [263] to learn the first-order asymptotically stable DS $\boldsymbol{f}_g(\boldsymbol{x})$, required for computing VSDS. To conclude, we conducted a total of 3×3×2 experiments on the robot.

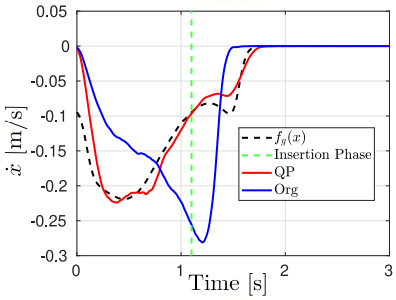
The results of the experiments are shown in Fig. 6.23 and 6.24. The first row in Fig. 6.23 shows the spatial motion in the constant stiffness case[4] overlaid on the

[4]The spatial motion in the varying stiffness case looks exactly the same, and therefore we omit it for brevity

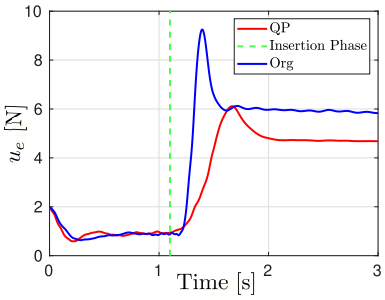
(a) Robot Setup



(b) Spatial Position



(c) Desired vs actual velocity



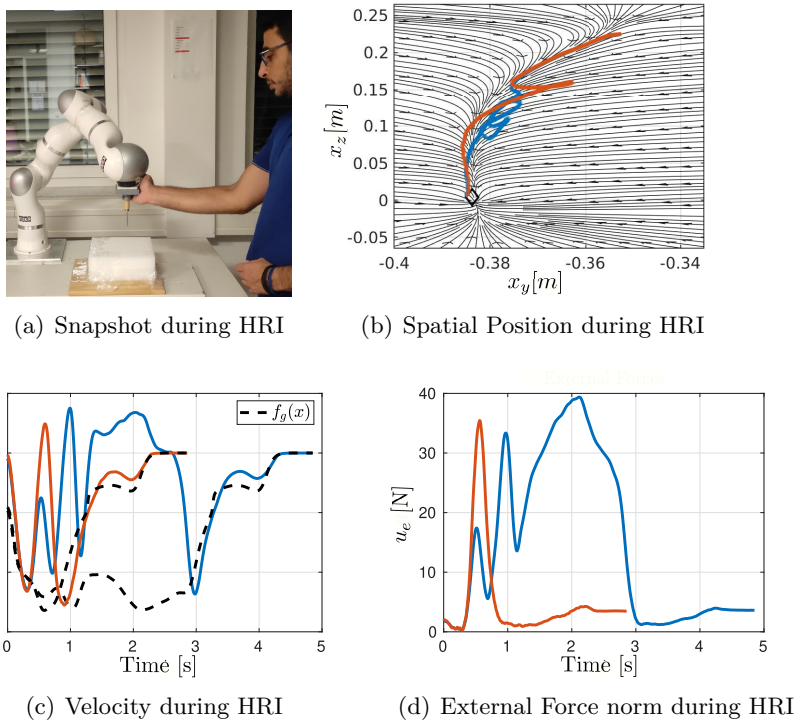
(d) External Force norm

Fig. 6.26: Experimental results of our drilling-like task, with the first figure showing the experimental setup of the robot, while the last three figures show the results of the task execution for the QP (red) and the Org.(blue) VSDS approaches. ©2023 IEEE [20].

streamlines of the original VSDS. The second row shows the actual robot velocity compared to the desired one computed by $\boldsymbol{f}_g(\boldsymbol{x})$ for the constant stiffness case. In Fig. 6.24(a), we show the mean and standard deviation of the Root Mean Square (RMS) velocity error ($\dot{\boldsymbol{e}}$) for the three controllers over all the executed motions (6 for each controller). Finally, Fig. 6.24(b) shows an example of the tank state from the same condition for the three control formulations.
For the original VSDS, the actual robot velocity is clearly different from the desired velocity profile $\boldsymbol{f}_g(\boldsymbol{x})$. This gets resolved by adding the feedforward term $\boldsymbol{f}_f(\boldsymbol{x})$, which improves the tracking accuracy of the reference velocity, and where the VF approach yields best tracking results, reflected by the lowest mean for the RMS error. On the other hand, all three controllers are able to guide the robot to the global attractor, with the tank rapidly rapidly converging to zero close to the equilibrium, which is also consistent with the simulation results of sec. 6.4.2.

Safety

In the second experiment, we validate the safety of the presented approaches in unexpected collisions, by placing a carton box in the path of the planned robot motion, as shown in Fig. 6.25(a). We learned a first-order DS based on a straight line



(a) Snapshot during HRI

(b) Spatial Position during HRI

(c) Velocity during HRI

(d) External Force norm during HRI

Fig. 6.27: Experimental Results of the drilling task for the QP approach where the robot is subjected to perturbations applied by a human. The first figure shows a snapshot where a human physically interacts with the robot, while the remaining figures show the spatial position, velocity and external forces from two different HRI experiments, depicted in orange and blue. ©2023 IEEE [20].

minimum jerk trajectory in the y- direction, and use it to construct our VSDS, where we also compare the same three VSDS variations from the previous subsection. The results of this experiment are shown in Fig. 6.25. Clearly, the interaction is safe for the original VSDS and the QP cases, highlighted by the relatively low external force (Fig. 6.25(f)), with a slightly higher force for the QP case. Note also the fact that the robot does not "push" the carton box, which can be verified by the steady state robot position (Fig. 6.25(e)) for these two cases is right at the collision point with the box (Fig. 6.25(b), 6.25(c)). On the contrary, the robot keeps on moving against the box (Fig. 6.25(d)) for the VF case, which results in a much higher collision force. This effect can be mainly attributed to the feed-forward term in equation (6.27), where increasing damping gains to a certain extent improves the velocity tracking, however at the expense of a higher steady state control force, which eventually increases the collision force. We would like to note however that while the magnitude steady state external force in the VF case was close to the external force we noticed in section 6.2.3 in the case of a time-indexed trajectory, the VF controller is still safer in the sense the control force does not increase over time, which typically results in

aggressive robot motions once the obstacle blocking the robot is removed.

Interaction Task

In this experiment, we tested our approach in a simplified drilling-like task which requires the robot to penetrate a foam surface with a needle-like tool mounted on its end-effector (Fig. 6.26(a)). For such a task, the robot starts from an initial position above the surface, approaches the drilling point with an arbitrary velocity and ideally maintains a constant low velocity during the insertion phase and therefore, following a specific velocity profile would be desirable. A human provides demonstrations to the robot in gravity compensation mode, while recording the end-effector position, and obtaining the velocities via finite differences, which serve as training data to learn a first-order DS with SEDS [13]. This is then used to construct our VSDS, where we use a state-varying stiffness profile that starts with a constant stiffness of 1000 N/m, and increases smoothly to 1800 N/m with a minimum jerk trajectory slightly before approaching the insertion location to compensate for the physical interaction. We compare the performance of our original VSDS approach, with the QP approach for the design of the $\boldsymbol{f}_f(\boldsymbol{x})$. As can be shown in Fig. 6.26, the task can be completed with both approaches. Note for the QP approach, the actual robot velocity follows well the desired velocity $\boldsymbol{f}_g(\boldsymbol{x})$ (Fig. 6.26(c))[5], which also reflects the human strategy used during the demonstrations to maintain a constant low velocity during the insertion phase. This results in a smooth task execution, as compared to the original VSDS approach, where the robot has a velocity profile that correlates with the stiffness, increasing during the insertion phase. This leads to a larger impact and in a consequence a higher overshoot in the external force sensed at the robot end-effector can be observed, as compared to the QP approach (Fig. 6.26(d)). In the second set of experiments (Fig. 6.27), we compare the robustness of our QP control approach to perturbations, applied by a human physically interacting with the robot during task execution. The robot reacts in a safe and compliant manner to the applied disturbances, and is able to resume smoothly the task execution, while still maintaining the desired velocity profile during the insertion phase (Fig. 6.27(c)).

6.5 Orientation Control with VSDS

In this section, we extend our VSDS framework to handle orientations, thereby obtaining a control algorithm that simultaneously shapes the robot orientational motion and rotational impedance, in a closed-loop manner that is always aware of the current robot orientation. Our approach relies on Unit Quaternions (UQ), a singularity-free representation of orientations. However, this poses a challenge since UQ geometric constraints (i.e., unitary norm) must be respected in the control formulation. This is a key difference with respect to position trajectories, which can be modeled in a decoupled manner. We overcome this problem by relying on the

[5]For clarity, we show only the velocity results in the main direction of motion which is the $z-$ axis.

Lie group formulation and leverage the mappings between the unit sphere and the tangent space to respect the underlying structure of UQ.

Unit quaternion group

A quaternion is an element of $\mathbb{H}$, where $\mathbb{H}$ is the space of quaternions and it is isomorph to $\mathbb{R}^4$. A quaternion can be defined using a hyper-complex number as $\boldsymbol{\theta} = \nu + \mathbf{u} \ : \ \nu \in \mathbb{R}$, $\mathbf{u} = [u_x, u_y, u_z]^\top \in \mathbb{R}^3$. A UQ $\boldsymbol{\theta} \in \mathcal{S}^3$ is an element of $\mathcal{S}^3$, i.e., the unit sphere embedded in $\mathbb{R}^4$. A UQ is a quaternion with a unit norm and can be used to describe an orientation in 3D space, where $\boldsymbol{\theta}$ and $-\boldsymbol{\theta}$ represent the same orientation. Quaternion norm is obtained by $||\boldsymbol{\theta}|| = \sqrt{\nu^2 + u_x^2 + u_y^2 + u_z^2}$. The conjugation of a quaternion $\boldsymbol{\theta}$ is defined as $\bar{\boldsymbol{\theta}} = \nu + (-\mathbf{u})$, while the multiplications of $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{S}^3$ is defined as

$$\boldsymbol{\theta}_1 * \boldsymbol{\theta}_2 = (\nu_1\nu_2 - \mathbf{u}_1{}^\top \mathbf{u}_2) + (\nu_1\mathbf{u}_2 + \nu_2\mathbf{u}_1 + \mathbf{u}_1 \times \mathbf{u}_2).$$

We project UQ back and forth between the unit sphere manifold $\mathcal{S}^3$ and the tangent space $\mathbb{R}^3$ by using exponential and logarithmic mapping operators. $\boldsymbol{\zeta} = \mathrm{Log}_{\boldsymbol{\theta}_2}(\boldsymbol{\theta}_1) : \mathcal{S}^3 \mapsto \mathcal{T}_{\boldsymbol{\theta}_2}\mathcal{S}^3$; $\mathcal{T}_{\boldsymbol{\theta}_2}\mathcal{S}^3 \equiv \mathbb{R}^3$, maps $\boldsymbol{\theta}_2 \in \mathcal{S}^3$ to $\boldsymbol{\zeta} \in \mathbb{R}^3$ wrt to $\boldsymbol{\theta}_1$. Consider $\boldsymbol{\theta} = \boldsymbol{\theta}_2 * \bar{\boldsymbol{\theta}}_1$,

$$\mathrm{Log}_{\boldsymbol{\theta}_1}(\boldsymbol{\theta}_2) = \mathrm{Log}(\boldsymbol{\theta}) = \begin{cases} \texttt{arccos}(\nu)\frac{\mathbf{u}}{||\mathbf{u}||}, & ||\mathbf{u}|| \neq 0 \\ [0\ 0\ 0]^\top, & \text{otherwise} \end{cases} \tag{6.49}$$

This mapping can be also used to define a distance metric upon $\mathcal{S}^3$ [264]

$$\mathrm{d}(\mathbf{q}_1, \mathbf{q}_2) = \begin{cases} 2\pi, & \boldsymbol{\theta}_1 * \bar{\boldsymbol{\theta}}_2 = \left[-1 + [0\ 0\ 0]^\top\right] \\ 2\|\mathrm{Log}_{\boldsymbol{\theta}_1}(\boldsymbol{\theta}_2)\| & \text{otherwise} \end{cases}. \tag{6.50}$$

Inversely, $\boldsymbol{\theta} = \mathrm{Exp}_{\boldsymbol{\theta}_1}(\boldsymbol{\zeta}) : \mathbb{R}^3 \mapsto \mathcal{S}^3$ maps $\boldsymbol{\zeta} \in \mathbb{R}^3$ to $\boldsymbol{\theta} \in \mathcal{S}^3$ so that it lies on the geodesic starting point from $\boldsymbol{\theta}_1$ in the direction of $\boldsymbol{\zeta}$

$$\mathrm{Exp}_{\boldsymbol{\theta}_1}(\boldsymbol{\zeta}) = \begin{cases} \left[\cos(||\boldsymbol{\zeta}||) + \sin(||\boldsymbol{\zeta}||)\frac{\boldsymbol{\zeta}}{||\boldsymbol{\zeta}||}\right] * \boldsymbol{\theta}_1, & ||\boldsymbol{\zeta}|| \neq 0 \\ \left[1 + [0\ 0\ 0]^\top\right] * \boldsymbol{\theta}_1, & \text{otherwise} \end{cases} \tag{6.51}$$

It is worth mentioning that UQ are often computed numerically from rotation matrices, for instance when robot's end-effector poses are collected via kinesthetic teaching. In these cases, it may happen that the algorithm returns a quaternion at step t and an antipodal quaternion at $t+1$. To ensure that the demonstration of an orientation profile is discontinuity-free, we can check that the dot product between each adjacent UQ is greater than zero. Otherwise, we flip $\boldsymbol{\theta}_{t+1}$ such as $\boldsymbol{\theta}_{t+1} = -\boldsymbol{\theta}_{t+1}$.

Remark 1 As discussed in [265], the domain of $\mathrm{Log}(\cdot)$ extends to all $\mathcal{S}^3$ except $-1 + [0\ 0\ 0]^\top$, while the domain of $\mathrm{Exp}(\cdot)$ is constrained by $\|\boldsymbol{\zeta}\| < \pi$. Restricting the domain to $\|\boldsymbol{\zeta}\| < \pi$ makes (6.49) and (6.51) bijective.

6.5.1 Formulation based on Unit Quaternions

In this section, we formulate the VSDS algorithm for the orientation case. Similar to the previous sections, we assume that there exists a known nominal motion plan represented by a first-order dynamical system $\boldsymbol{\omega}_d = \boldsymbol{f}_g(\boldsymbol{\theta})$ asymptotically stable around a global equilibrium $\boldsymbol{\theta}^*$, with $\boldsymbol{f}_g(\cdot) : \mathcal{S}^3 \mapsto \mathbb{R}^3$ a continously differentiable function that maps a UQ $\boldsymbol{\theta} \in \mathcal{S}^3$ into a desired angluar velocity $\boldsymbol{\omega}_d \in \mathbb{R}^3$. As shown in [266, 267], the function $\boldsymbol{f}_g(\cdot)$ can be learned from demonstration by projecting quaternion trajectories in the tangent space using the logarithmic map (6.49). We also have a desired, possibly state-varying stiffness profile $\boldsymbol{K}_o(\boldsymbol{\theta}) \in \mathbb{R}^{3\times3}$ which describes the desired interaction behavior for the orientation degrees of freedom.

In the following, we outline the key elements of the VSDS algorithm for orientations represented as UQs. As discussed earlier, applying operations from Euclidean geometry directly on UQ leads to inaccuracies. Therefore, we make consistent use of projections back and forth between the manifold and the tangent space, since elements of the tangent space have a euclidean geometry, and accordingly we can exploit the well-known linear algebra and arithmetic tools to manipulate its elements.

Via Points Extraction

The first step in our VSDS algorithm is to obtain a sequence of N via-points generated from one of the integral curves of $\boldsymbol{f}_g(\boldsymbol{\theta})$ starting at an initial orientation $\boldsymbol{\theta}_0$. These via-points act as attractors for the local springs, thereby shaping the motion, and are chosen to be equidistant to ensure a smooth velocity profile. The via-points extraction process is detailed in Algorithm 5, where first the motion plan DS is simulated with a sampling time Δt until reaching $\boldsymbol{\theta}^*$, generating a temporary sequence of via-points, which are then projected in the tangent space at the goal quaternion $\boldsymbol{\theta}^*$ (Lines 7-13). The temporary via-points are then re-sampled to obtain N equidistant via-points $\boldsymbol{\zeta}_{l,0}, \boldsymbol{\zeta}_{l,i}, \dots \boldsymbol{\zeta}_{l,N}$ in the tangent space and N equidistant via-points $\boldsymbol{\theta}_{l,0}, \boldsymbol{\theta}_{l,i}, \dots \boldsymbol{\theta}_{l,N}$ in the unit quaternion space (Lines 15-22).

Local Springs

The goal now is to construct N local spring actions around the generated via-points which serve as local attractors for each spring. The spring actions are realized by a DS represented in the tangent space, such that

$$\boldsymbol{f}_{l,i}(\boldsymbol{\zeta}) = \boldsymbol{A}_{l,i}\mathrm{Log}_{\boldsymbol{\theta}_{l,i}}(\boldsymbol{\theta}) \tag{6.52}$$

where $\mathrm{Log}_{\boldsymbol{\theta}_{l,i}}(\boldsymbol{\theta})$ is the projection of the current robot orientation $\boldsymbol{\theta}$ in the tangent space at the via-point $\boldsymbol{\theta}_{l,i}$ (and it plays the role of an error term). Note that we write $\boldsymbol{f}_{l,i}(\boldsymbol{\zeta})$ to highlight the fact that $\boldsymbol{f}_{l,i}$ maps elements that belong to the tangent space. The stiffness of each spring is $\boldsymbol{A}_{l,i}$, which is constructed based on the desired stiffness profile $\boldsymbol{K}_o(\boldsymbol{\theta})$, and designed as

$$\boldsymbol{A}_{l,i} = \boldsymbol{Q}_{l,i}\boldsymbol{K}_{o,i}\boldsymbol{Q}_{l,i}^T, \tag{6.53}$$

Algorithm 5: SampleViaPoints

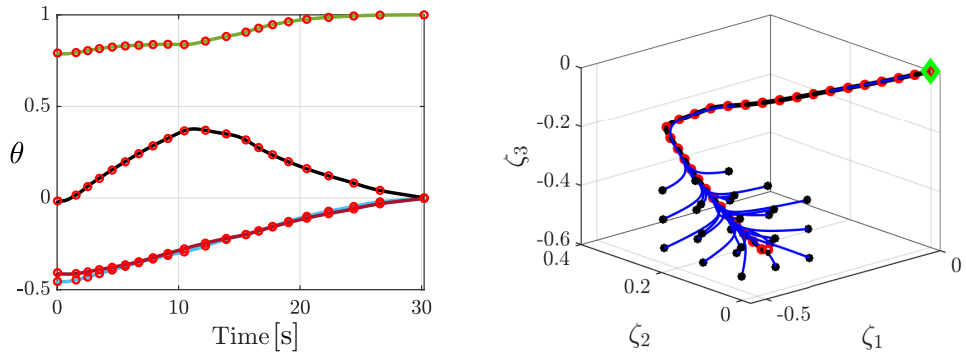
input : Motion plan DS $\boldsymbol{f}_g$, initial $\boldsymbol{\theta}_0$ and goal orientation $\boldsymbol{\theta}^*$, Number of via-points N, sampling time Δt

$n = 0$;
$k = 1$;
$d_{sum} = 0$;
$\boldsymbol{\zeta}_{l,n} = \mathrm{Log}_{\boldsymbol{q}^*}(\boldsymbol{q}_0)$;
$\boldsymbol{\zeta}_{tmp,k} = \mathrm{Log}_{\boldsymbol{q}^*}(\boldsymbol{q}_0)$;
$\boldsymbol{q}_{tmp,k} = \boldsymbol{q}_0$;
while $\mathrm{d}(\boldsymbol{q}_k, \boldsymbol{q}^*) > \epsilon$ **do**
  $k = k + 1$;
  $\boldsymbol{\zeta}_{tmp,k} = \mathrm{Log}_{\boldsymbol{\theta}^*}(\boldsymbol{\theta}_{tmp,k})$;
  $\boldsymbol{\omega}_{tmp,k} = \boldsymbol{f}_g(\boldsymbol{\theta}_{tmp,k})$;
  $\boldsymbol{\theta}_{tmp,k+1} = \mathrm{Exp}_{\boldsymbol{\theta}_{tmp,k}}(\frac{\Delta t}{2}\boldsymbol{\omega}_{d,k})$;
  $d_{sum} = d_{sum} + \mathrm{d}(\boldsymbol{\theta}_{tmp,k}, \boldsymbol{\theta}_{tmp,k-1})$;
end
$d_l = d_{sum}/N$;
for $i \leftarrow 1$ **to** k **do**
  if $\mathrm{d}(\boldsymbol{\theta}_{tmp,i}, \boldsymbol{\theta}_{k,n}) \geq d_l$ **then**
    $n = n + 1$;
    $\boldsymbol{\theta}_{l,n} = \boldsymbol{\theta}_{tmp,i}$;
    $\boldsymbol{\zeta}_{l,n} = \boldsymbol{\zeta}_{tmp,i}$;
    $\boldsymbol{\omega}_{l,n} = \boldsymbol{\omega}_{tmp,i}$;
  end
end
$\boldsymbol{\zeta}_{l,N} = \mathrm{Log}_{\boldsymbol{\theta}^*}(\boldsymbol{\theta}^*) = [0\,0\,0]^\top$, $\boldsymbol{\theta}_{l,N} = \boldsymbol{\theta}^*$;

where $\boldsymbol{K}_{o,i} = \boldsymbol{K}_o(\boldsymbol{\theta}_{l,i})$ is the value of the stiffness matrix evaluated at the current via-point $\boldsymbol{\theta}_{l,i}$, while $\boldsymbol{Q}_{l,i}$ is a matrix constructed to have its first eigen vector normalized and pointing in the direction of motion, i.e., $\frac{\zeta_{l,i}}{||\zeta_{l,i}||}$ whereas the remaining vectors are derived to be perpendicular to it, via a Gram-Schmit orthogonalization procedure. Once the springs are constructed, we combine them via a non-linear weighted sum. The weights act as a state-dependent activation magnifying/reducing the effect of each spring depending on the region of the state space, while achieving a smooth transition. As in the original VSDS, we generate this smooth transition using a Gaussian kernel. A Gaussian kernel on the unit sphere can be defined as [268]

$$w_i(\boldsymbol{\zeta}) = \exp\left(-\frac{\mathrm{Log}_{\boldsymbol{\theta}_{cen,i}}(\boldsymbol{\theta})^\top \mathrm{Log}_{\boldsymbol{\theta}_{cen,i}}(\boldsymbol{\theta})}{2\left(\sigma^i\right)^2}\right) \tag{6.54}$$

where $\boldsymbol{\theta}_{cen,i}$ denotes the center of the i_{th} DS, i.e., the mean of $\boldsymbol{\theta}_{l,i}$ and $\boldsymbol{\theta}_{l,i-1}$, and it can be efficiently computed using the EM-based approach outlined in [268].



(a) Quaternion trajectory (b) Tangent space trajectories

Fig. 6.28: Left: reference path for VSDS as a UQ where the 4 lines correspond to the v, u_x, u_y and u_z. Right: shows the state trajectories of a second-order DS driven by VSDS starting from multiple initial orientations (black dots). The black line is the reference path, while the green rhombus is the goal. In both plots, the red dots are the computed via-points.

$\sigma^i \in \mathbb{R}^+$ is proportional to the distance between two tangent space via points, computed using (6.50). The actual spring weights used are then the normalized kerenels $\tilde{w}_i(\boldsymbol{\zeta}) = \frac{w_i(\boldsymbol{\zeta})}{\sum_{j=1}^{N} w_j(\boldsymbol{\zeta})}$.

Control policy

Finally, our VSDS control policy is devised as the weighted combination of all local springs, such that

$$\boldsymbol{\tau}_{vs}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \tilde{w}_i\left(\boldsymbol{\zeta}\right) \boldsymbol{f}_{l,i}\left(\boldsymbol{\zeta}\right) \tag{6.55}$$

which represents a non-linear torque field that maps the current orientation $\boldsymbol{\theta}$ (computed through $\boldsymbol{\zeta}$) into a torque $\boldsymbol{\tau}_{vs} \in \mathbb{R}^3$ that simultaneously shapes the robot motion and rotational stiffness, in closed loop. Fig. 6.28(a) shows the computed via-points (red-dots) along a quaternion trajectory simulated from a first-order DS learnt with [267], while Fig. 6.28(b) visualizes the same DS in the tangent space at the goal $\boldsymbol{\theta}^*$ i.e $\mathcal{T}_{\boldsymbol{\theta}^*}\mathcal{S}^3$, and where also depicted the evolution of a rotational second order dynamical system driven by VSDS starting from multiple initial positions. Please note that all trajectories converge to the reference path along which the via-points are computed, thanks to the symmetric attraction behavior encoded within VSDS. The entire procedure for the VSDS approach including the initialization and the real time control is outlined in Algorithm 6.
The Cartesian control torque for the orientation degrees of freedom can then be set as

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{vs} - \boldsymbol{D}_o\boldsymbol{\omega} \tag{6.56}$$

where $\boldsymbol{\omega}$ is the actual angular velocity of the robot, while $\boldsymbol{D}_o$ is a damping matrix. To command the robot joints, the actuation torques sent to the motors are computed

Algorithm 6: Orientation Control with VSDS

input : $\boldsymbol{f}_g$, $\boldsymbol{\theta}^*$, $\boldsymbol{\theta}_0$, Δt, N, Current robot orientation $\boldsymbol{\theta}$ and Desired stiffness profile $\boldsymbol{K}_o(\boldsymbol{\theta})$

VSDS Initialization:
Algorithm 1 ;
for $i \leftarrow 1$ **to** N **do**
  $\boldsymbol{K}_{o,i} = \boldsymbol{K}_o(\boldsymbol{\theta}_{l,i})$;
  $\boldsymbol{A}_{l,i} = \boldsymbol{\theta}_i \boldsymbol{K}_{o,i} \boldsymbol{\theta}_i^T$;
  $\boldsymbol{\theta}_{cen,i}$ mean of $\boldsymbol{\theta}_{l,i}$ and $\boldsymbol{\theta}_{l,i-1}$ [268];
  $l_i = \mathrm{d}\left(\boldsymbol{\theta}_{l,i}, \boldsymbol{\theta}_{l,i-1}\right)$;
  $\sigma_i = \delta l_i$;
end
Control Loop:
while $\mathrm{d}(\boldsymbol{\theta}, \boldsymbol{\theta}^*) > \epsilon$ **do**
  for $i \leftarrow 1$ **to** N **do**
    $w_i\left(\boldsymbol{\zeta}\right) = \exp\left(-\frac{\mathrm{Log}_{\boldsymbol{\theta}_{cen,i}}(\boldsymbol{\theta})^\top \mathrm{Log}_{\boldsymbol{\theta}_{cen,i}}(\boldsymbol{\theta})}{2(\sigma^i)^2}\right)$;
  end
  for $i \leftarrow 1$ **to** N **do**
    $\boldsymbol{f}_{l,i}\left(\boldsymbol{\zeta}\right) = \boldsymbol{A}_{l,i} \mathrm{Log}_{\boldsymbol{\theta}_{l,i}}(\boldsymbol{\theta})$;
    $\tilde{w}_i\left(\boldsymbol{\zeta}\right) = \frac{w_i}{\sum_{j=1}^{N} \omega_j(\boldsymbol{\zeta})}$;
  end
  $\boldsymbol{\tau}_{vs}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \tilde{w}_i\left(\boldsymbol{\zeta}\right) \boldsymbol{f}_{l,i}\left(\boldsymbol{\zeta}\right)$;
end

as

$$\boldsymbol{u}_c = \boldsymbol{J}^\top \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{\tau} \end{bmatrix}, \tag{6.57}$$

with $\boldsymbol{J}$ as the end-effector jacobian, and where the robot is assumed to be pre-gravity compensated. As for $\boldsymbol{F} \in \mathbb{R}^3$, it is a generic control force for the translational degrees of freedom, which can be commanded also via VSDS, or with a standard impedance controller.

6.5.2 Experimental Validation

In this section, we aim to validate our approach in a series of experiments on a 7 DOF KUKA LWR robot. We implemented our control algorithm on a standard core i7 PC with a 16 GB RAM, in C++. The robot was controlled via the Fast Research Interface in the joint impedance control mode, where the feed forward torque command is used for our VSDS control law. The frequency of the control loop runs at 500 Hz. In the following, we present a series of experiments that highlight the ability of our VSDS to follow complex orientation profiles, safety during disturbances

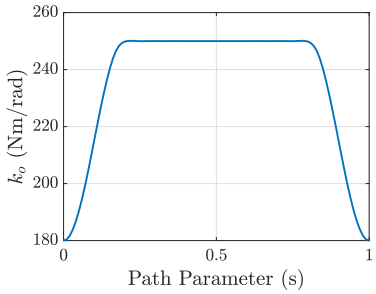
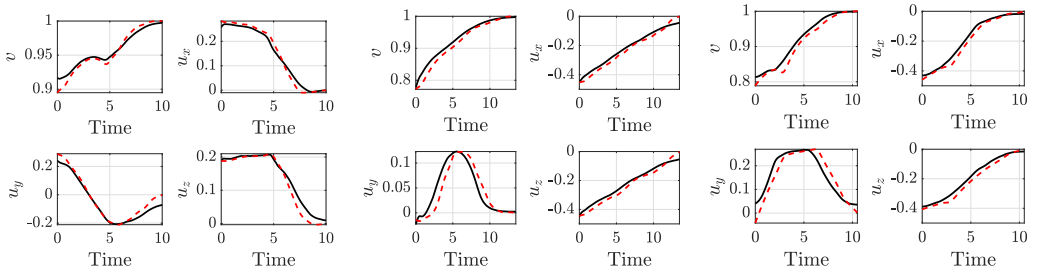
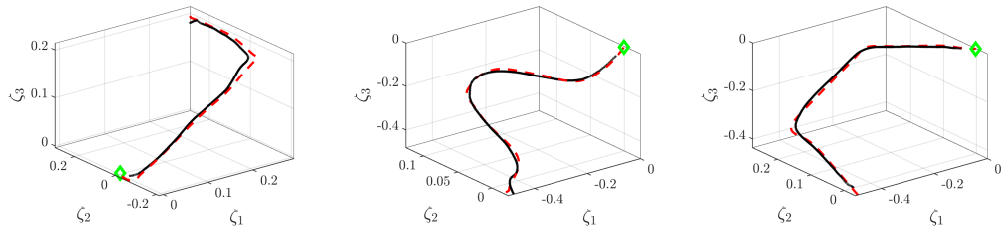


Fig. 6.29: Variable Stiffness Profile



(a) JShape: Quat. Trajectory (b) Worm: Quat. Trajectory (c) Trapezoid: Quat. Trajectory



(d) JShape: Tangent Motion (e) Worm: Tangent Motion (f) Trapezoid: Tangent Motion

Fig. 6.30: Experimental results for free motion execution, where the black line depicts the actual robot motion, while the dotted red is the desired one.

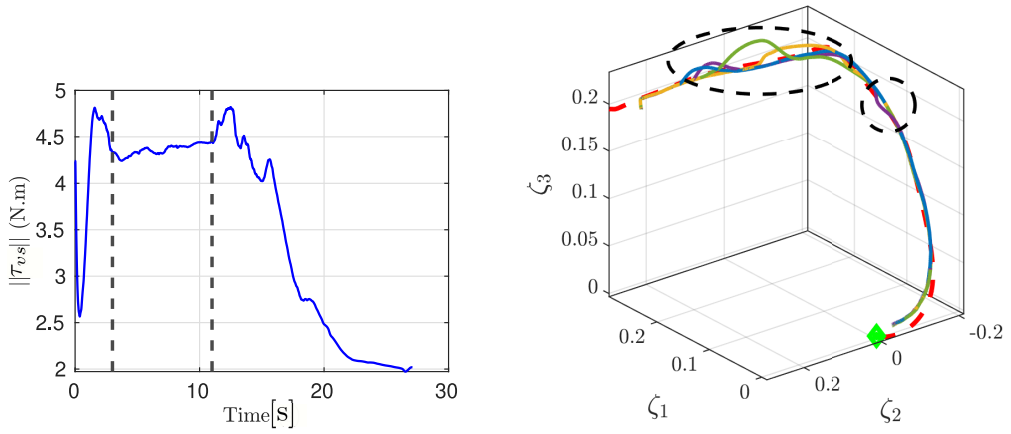
and finally, an interaction task with varying stiffness levels. For all experiments, we used $N = 30$ as the number of local spring actions. In order to generate a first order DS, we used the Riemanian manifold stable DS (SDS-RM) [267] to learn the orientation motion plan DS $\boldsymbol{f}_g$ from demonstrations. The idea of SDS-RM is to encode demonstrations that belong to a Riemannian manifold into a first order stable DS. This is achieved by learning a diffeomorphism; a bijective smooth mapping with a continuous inverse; between tangent spaces to map a globally asymptotically stable base DS into a DS that can model complex motions, while preserving the stability properties of the base DS. Since UQ belong to Riemannian manifolds, SDS-RM can be adequately deployed to model demonstrated orientation trajectories.

Free Motion Evaluation

In this subsection, we test the ability of our control law to follow a desired orientation path. To that end, we use the Riemannian LASA Data Set [266, 267] to obtain orientation demonstration data from three shapes: JShape, Worm and a Trapezoid. We then use SDS-RM [267] to learn a first-order DS, which is subsequently used to generate the via points for VSDS. For the stiffness, we used a variable stiffness profile parameterized as a function of the path parameter $s \in [0, 1]$ such that $s = 0$ at $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ and $s = 1$ at $\boldsymbol{\theta} = \boldsymbol{\theta}^*$. The stiffness profile is designed to start with a stiffness of 180 Nm/rad, rise smoothly to 250 Nm/rad and fall again back to 180 Nm/rad toward the end of the motion, as shown in Fig. 6.29, such that $\boldsymbol{K}_o = k_o(s)\boldsymbol{I}_3$. The translational degrees of freedom of the robot were commanded with a cartesian impedance controller to hold the initial position.
The results of this experiment are shown in Figure 6.30, where depicted the actual robot (solid black) and desired (dotted red) orientation motions for the three shapes. The 3D plots in Figures 6.30(d), 6.30(e) and 6.30(f) highlight the evolution of the orientation trajectory projected in the tangent space $\mathcal{T}_{\boldsymbol{\theta}^*}\mathcal{S}^3$, while Figures 6.30(a), 6.30(b) and 6.30(c) show the evolution of the scalar and vector parts of the quaternion trajectories. The desired motion is generated through the open-loop integration of the learnt first order DS $\boldsymbol{f}_g(\boldsymbol{\theta})$, which represents the nominal motion plan. As highlighted in the figure, the actual robot orientation follows well the desired orientation profile. Please note, to ease visualization, the desired quaternion trajectory generated by the first order DS is scaled in time, in order to make it in the same time scale of the actual one.

Safety

We further validate the safety and robustness of our control algorithm by testing its abilities to cope with various disturbances. The robot has to execute the J-Shape motion in a similar setting to the previous subsection, however, with a human applying perturbations by physically interacting with the robot. In the first case, the human attempts to stop the robot motion, simulating a potential collision or an obstacle that might block the robot. The robot reacts in a compliant manner and resumes the task smoothly once the human releases the robot. This is also reflected in the VSDS control torques norm shown in figure 6.31(a), which remains almost constant during the period the human is holding the robot, highlighted by the two dotted vertical lines. This smooth behavior can be realized thanks to the closed-loop configuration, where the controller/motion generator are encoded as a single entity, that is always aware of the current robot state. On the other hand, a time-driven controller will result in a control torque that progressively increases with time, resulting in an abrupt and potentially dangerous behavior once the obstacle is removed.
In the second case, the human repeatedly pushes the robot attempting to deviate it from its desired path. The robot again reacts to the applied disturbances compliantly, resuming the planned task by attracting back to the desired orientation path, as



(a) VSDS Control Torque

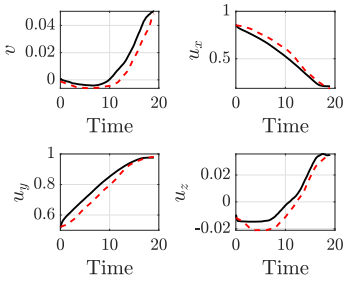
(b) Tangent Space trajectories

Fig. 6.31: Left: norm of the VSDS control torque during an execution where the human attempts to block the robot motion, where the black lines depict the duration of the interaction. Right: quaternion trajectories from several experiments where the human applies random disturbances to the robot. The dotted red is the reference path in the tangent space, while the dotted ellipsoids highlight the instances where the human applied the disturbances.

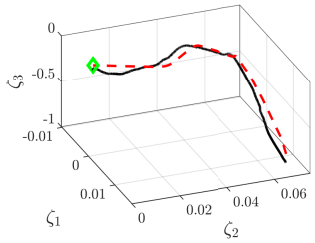
shown in the robot motions visualized in tangent space (Figure 6.31(b)). This can be fulfilled thanks to the symmetric attraction behavior of VSDS, which is also consistent with the simulation results of Fig. 6.28(b).

Interaction Task

In this experiment, we validate the ability of our approach to execute physical interaction tasks. In particular, we chose a cutting task where a scalpel was mounted on the robot end-effector, and where the goal is to perform a curve-shaped cutting motion on air-drying modeling clay that naturally hardens over time. To achieve this objective, the orientation of the robot must continuously change in order to keep the cutting edge of the scalpel always aligned with the current motion direction. To learn the task, a human provides one demonstration to the robot via kineasthetic teaching with the robot in gravity compensation mode where the robot pose data is collected. The recorded orientations are then used to learn a first order DS with SDS-RM, which is used to generate the via-points for VSDS. During execution, we perform the same cutting motion in two different settings: In the first case, the robot has to cut a freshly opened piece of soft clay, and therefore we use a relatively low constant stiffness of $\boldsymbol{K}_o = 130\boldsymbol{I}_3$ Nm/rad to perform the cut. In the second case, we use a stiffness of $\boldsymbol{K}_o = 200\boldsymbol{I}_3$ Nm/rad to cut an older and comparatively harder material. For the translational motion, we simply play-back the recorded trajectory



(a) Cutting: Quat. Trajectory

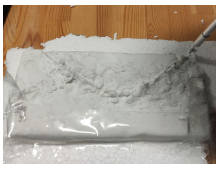


(b) Cutting: Tangent Motion

(c) Soft before cut

(d) Soft after cut

(e) Hard before cut



(f) Hard after cut

Fig. 6.32: Experimental results for Cutting

and track it with a Cartesian impedance controller. As shown in Figure 6.32[6], the robot adapts its orientation based on the desired profile throughout the motion in order to prevent any lateral knife motions, and completes the task successfully.

6.6 Summary

In this chapter, we proposed an approach called Variable Stiffness Dynamical Systems (VSDS). The proposed formulation takes as input a first order DS that describes a reference motion plan, a desired user-defined constant/state varying stiffness profile, and generates a new controller that simultaneously modulates the robot motion and its interaction behavior. The control policy works in a closed-loop manner based on the actual robot state, endowing it with enhanced reactivity and robustness in the face of perturbations or accidental collisions. Our proposed controller is built as a set of non-linearly weighted springs centered around a number of via-points generated by a nominal DS motion plan. We tested our approach in real robot experiments showcasing the ability to successfully perform contact tasks in autonomous settings, as well as its safety. We also showed an application for VSDS in shared control, where we exploited several VSDS features to realize haptic guidance, authority allocation and incremental learning. The shared control approach was also validated in a user study, where we also compared our VSDS controller for haptic guidance to several state-of-the-art controllers.
Subsequently, we extended the VSDS formulation to advance its potential in two

[6]We show only the orientation trajectories of the hard material, since the trajectories of the soft are similar.

main directions. First, We showed how to guarantee the stability of our controller by exploiting energy tanks to enforce passivity, thereby ensuring stable interactions with passive environments. We further extended our proof to include asymptotic stability with respect to a global attractor in free motion. Our second goal was to make our controller more suited for tasks that require trajectory tracking, while still benefiting from the inherit safety properties of the closed-loop configuration, as shown in our collision experiments. To achieve these goals, we proposed two formulations based on velocity error feedback and QP optimization of feedforward control terms.
Finally, we extended the VSDS approach for orientation tasks, to enable a robot to follow a desired orientation motion plan described by a first-order DS, with a user-defined rotational stiffness profile, in a closed-loop configuration. Our algorithm relies on UQ to represent orientations and projections between the manifold and tangent space, to realize a control law formulated as a non-linear weighted sum of local springs centered around local attractors in the tangent space. Thanks to the chosen formulation, similar to the translation case, the controller also realizes a spring-like attraction behavior around an orientation reference path, where the robot is pulled back to the path when perturbed. We validated during real-world experiments the ability of the proposed approach to follow complex orientation profiles, ensure safe and compliant reactions to perturbations, and in executing contact tasks that require continuous changes in orientation.

CHAPTER 7

Conclusion

7.1 Summary

The integration of robots into our daily lives necessitates a fundamental transformation in the way we conceptualize robot control strategies. It is evident that the conventional approach of employing rigid position control, as commonly applied in industrial robotics, falls short of the mark. What is required are control algorithms with the capability to execute tasks safely and compliantly. To this end, the main impetus behind this research endeavor was to equip physically interacting robots with both safety and adaptability. In the following, we provide a brief overview of how these objectives were addressed in this thesis.

Chapter 3: Hierarchical control and VIC are powerful tools that allow to enhance the flexibility and adaptability of robotic manipulators. These features however come at the expense of violating the stability of the closed-loop system; a dangerous consequence especially if robots are predestined to operate in the vicinity of humans. In chapter 3, we developed a hierarchical variable impedance controller to achieve multiple prioritized objectives parameterized with a time varying stiffness and damping. Stability of the controller in interaction with passive environments was guaranteed by ensuring passivity enforced through energy tanks. Furthermore, an additional safety layer was designed to limit the maximum kinetic energy of the robot, by regulating the power flow exchanged between the robot and the environment. As shown in the experimental validation, in addition to task execution, the controller enhances the safety and robustness of the robot in face of perturbations and disturbances.

Chapter 4: Several works have demonstrated that incorporating variable impedance control into teleoperation allows to expand the interaction control capabilities of teleoperated robots, as compared to constant impedance settings. This was our motivation in chapter 4, however in contrast to previous works where typically additional hardware is required to convey the human impedance, we leveraged LfD to learn

a biomimetic stiffness adaptation policy, that shapes the remote robot impedance during physical interactions. As experimentally validated, our approach provides a suitable trade-off between safety and performance, when compared to constant high or low stiffness conditions.
Chapter 5: Herein, we extend the tele-impedance approach of Chapter 4, into a shared control architecture dedicated for contact tasks. Thereby, in addition to the automatic remote robot impedance adaptation, the learning model features information related to the desired motion, forces as well task frames. This information is utilized by virtual fixtures that guide the human along the nominal trajectory that optimally fulfills the task. A user study on two different contact tasks showed that haptic guidance indeed improves the user performance during task execution. Additionally, the smart allocation of the controlled DOF between the autonomy and manual control provides the best results in both objective and subjective metrics.
Chapter 6: In Classical robot motion control, the robot is driven along a time-dependent trajectory generated by a motion generator in an open-loop manner. Such a configuration lacks reactivity and can be problematic particularly in unstructured environments. To solve this problem, we proposed a framework where motion generation and variable impedance control are integrated into a single controller that receives continuous feedback from the current state. The controller proved its capability in autonomous task execution, shared control, as well the substantial safety improvement compared to the classical open-loop configuration. Furthermore, theoretical guarantees were derived to ensure the system stability during interactions, as well as convergence in free motions. Finally, we extended the controller to the orientation case, leveraging a geometry-aware formulation using Unit Quaternions and Lie group operations.

7.2 Future Works

7.2.1 Hierarchical Control

The controller formulated in Chapter 3 realizes a strict hierarchy for an arbitrary number of tasks, where the performance criterion for each task is specified by a time varying spring and a damper, around a virtual equilibrium. Two implicit assumptions were made in this regard: First, the natural inertial behavior on each hierarchy level is preserved such that inertia shaping is avoided. Second, that the virtual equilibrium for all tasks is constant i.e the controller is formulated for the regulation case. Therefore, to unlock the full potential of the controller, it would be interesting to extend the proposed approach to consider the case where the impedance behavior is specified with time-varying inertia matrix and equilibrium trajectories as well. Shaping the robot inertia can be useful for example in collaborative applications where it may be desired to render a lighter robot to the human. The works in [113] and [269] can serve as a good starting point towards this objective, in combination with the approach presented in this thesis. Extending the controller however to the tracking case might be less straightforward. Recent work in [270] proposed a hierarchical

(constant) impedance-based tracking controller, however the authors only considered the case where the total task dimension is lower than or equal than the dimension of the robot joint space. Therefore, more investigation needs to be done on how to relax this constraint, and how to accommodate time-varying impedance parameters. Another aspect we considered for this controller was to limit the maximum kinetic energy of the control action, for enhanced safety. Herein, we assumed that the maximum kinetic energy limit is known and provided by the user a priori. In the future, it should analyzed how to derive this limit based on realistic human injury data, along the same lines of [158] and [271]. For example, such a limit could be derived using the Safe Motion Unit developed in [272], which determines the maximum safe velocity for the current robot reflected mass. This would result in context-dependent varying limits on the maximum kinetic energy allowed in the system, which in turn would maximize the system performance.

7.2.2 Teleoperation and Shared Control

One of the major focus areas in this thesis was the development of variable impedance and shared control methods for robots that operate in teleoperated settings. The scope of application of these approaches however was limited to the translation case. It would be certainly compelling to expand the usability of these approaches to tasks that require the control of orientation as well e.g screwing or cutting along a curved path. To this end, the next logical step would be to extend these approaches to encompass orientation control, by leveraging singularity free representations such as Unit Quaternions and Lie group operations. Ongoing research aims at extending the approach presented in chapter 4 to the orientation case, starting from a rotational mass-spring-damper formulation to describe the task dynamics. Subsequently, the rotational stiffness can be related to input features such as the measured external torques at the end-effector through GMM. Once this model is established, it can be employed in a shared control framework similar to that of Chapter 5, and where rotational virtual fixtures guide the user along the nominal orientation trajectory.
Related to that, we see the approach developed in Chapter 5 as a first step in the direction of having a truly adaptive Shared control framework dedicated for contact tasks. Such a framework should assist the human on the trajectory level along the nominal task dynamics taking into account relevant task aspects such the motion, force and impedance (what we proposed in this thesis), but should be also augmented with a higher level of intelligence where a library of models for several tasks or skills is saved. Consequently, the framework should be equipped with tools to infer the current human intention with respect to the desired task that shall be executed, and generate the appropriate guidance accordingly. Additionally, such a framework would benefit from an authority allocation scheme in order to allow for the human to escape temporarily the guidance e.g to refine the execution of certain parts of the task.

7.2.3 Safe Motion Generation

We developed in chapter 6 a flexible framework for reactive motion generation and variable impedance control, which was validated in autonomous execution, shared control and was also extended to orientation control. Ongoing research efforts are dedicated towards extending the VSDS for the orientation case with trajectory tracking and a stability proof, as done for the translation case. Other than that, in the future, research should be dedicated towards finding other interesting applications for the VSDS controller. For example, while we showcased the benefits of the controller for a teleoperated shared control scenario, it would be interesting to test the performance of VSDS in a setting where both the human and the robot are collocated, such is the case in co-manipulation or collaborative tasks.
In addition to that, due to the flexibility of VSDS in encoding a wide array of motions and stiffness behaviors, as well as the safe nature of the controller, VSDS could be a powerful candidate to formulate the policy in Reinforcement learning applications. This would be somewhat similar to the approach developed in [109], but in contrast to the open-loop formulation used in [109], the closed-loop configuration of VSDS means a safer exploration scheme, with guaranteed bounded robot motions, thanks to the passivity proof of section 6.4.2.

List of Figures

List of Tables

Bibliography

[1] F. Abu-Dakka and M. Saveriano, "Variable impedance control and learning – a review," *Frontiers in robotics and AI*, vol. 7, 2020.

[2] E. Burdet, R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato, "The central nervous system stabilizes unstable dynamics by learning optimal impedance," *Nature*, vol. 414, no. 6862, pp. 446–449, 2001.

[3] Y. Li, N. Jarrassé, and E. Burdet, "Versatile interaction control and haptic identification in humans and robots," *Geometric and Numerical Foundations of Movements*, pp. 187–206, 2017.

[4] N. Hogan, "Impedance Control: An Approach to Manipulation: Part I—Theory," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–7, 1985.

[5] ——, "Impedance control: An approach to manipulation: Part II—implementation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 8–16, 1985.

[6] ——, "Impedance control: An approach to manipulation: Part III—applications," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 17–24, 1985.

[7] F. J. Abu-Dakka, L. Rozo, and D. G. Caldwell, "Force-based variable impedance learning for robotic manipulation," *Robotics and Autonomous Systems*, vol. 109, pp. 156 – 167, 2018.

[8] G. Ganesh, N. Jarrassé, S. Haddadin, A. Albu-Schaeffer, and E. Burdet, "A versatile biomimetic controller for contact tooling and haptic exploration," *IEEE International Conference on Robotics and Automation*, 2012.

[9] M. Li, H. Yin, K. Tahara, and A. Billard, "Learning object-level impedance control for robust grasping and dexterous manipulation," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6784–6791, 2014.

[10] L. D. Rozo, S. Calinon, D. Caldwell, P. Jiménez, and C. Torras, "Learning collaborative impedance-based robot behaviors," *AAAI Conference on Artificial Intelligence*, 2013.

[11] L. Peternel, T. Petrič, E. Oztop, and J. Babič, "Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach," *Autonomous robots*, vol. 36, pp. 123–136, 2014.

[12] A. Dietrich, X. Wu, K. Bussmann, C. Ott, A. Albu-Schäffer, and S. Stramigioli, "Passive hierarchical impedance control via energy tanks," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 522–529, 2017.

[13] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[14] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[15] M. Talha, E. A. M. Ghalamzan, C. Takahashi, J. Kuo, W. Ingamells, and R. Stolkin, "Towards robotic decommissioning of legacy nuclear plant: Results of human-factors experiments with tele-robotic manipulation, and a discussion of challenges and approaches for decommissioning," *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2016.

[16] A. Ajoudani, N. Tsagarakis, and A. Bicchi, "Tele-impedance: Teleoperation with impedance regulation using a body-machine interface," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1642–1656, 2012.

[17] Y. Michel, C. Ott, and D. Lee, "Safety-aware hierarchical passivity-based variable compliance control for redundant manipulators," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3899–3916, 2022.

[18] Y. Michel, R. Rahal, C. Pacchierotti, P. Robuffo Giordano, and D. Lee, "Bilateral teleoperation with adaptive impedance control for contact tasks," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 1–1, 2021.

[19] Y. Michel, Z. Lee, and D. Lee, "A learning-based shared control approach for contact task," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8002–8009, 2023.

[20] Y. Michel, M. Saveriano, and D. Lee, "A passivity-based approach for variable stiffness control with dynamical systems," *IEEE Transactions on Automation Science and Engineering*, 2023.

[21] H. Xue, Y. Michel, and D. Lee, "A shared control approach based on first-order dynamical systems and closed-loop variable stiffness control," *Journal of Intelligent and Robotic Systems*, vol. 109, no. 4, pp. 85–98, 2023.

[22] Y. Michel, C. Ott, and D. Lee, "Passivity-based variable impedance control for redundant manipulators," *IFAC-PapersOnLine*, pp. 9865–9872, 2020.

[23] X. Chen, Y. Michel, and D. Lee, "Closed-loop variable stiffness control of dynamical systems," *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 163–169, 2021.

[24] Y. Michel, M. Saveriano, F. J. Abu-Dakka, and D. Lee, "Orientation control with variable stiffness dynamical systems," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4457–4463, 2023.

[25] Y. Michel, M. Saveriano, and D. Lee, "A novel safety-aware energy tank formulation based on control barrier functions," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5206–5213, 2024.

[26] Y. Michel, Y. Abdelhalem, and G. Cheng, "Passivity-based teleoperation with variable rotational impedance control," *IEEE Robotics and Automation Letters*, 2024 (under review).

[27] Y. Michel, K. H. Schulleri, L. Johannsen, and D. Lee, "Coordination tending towards an anti-phase relationship determines greater sway reduction during entrainment with a simulated partner," *Human Movement Science*, vol. 89, p. 103090, 2023.

[28] H. Ahn, Y. Michel, T. Eiband, and D. Lee, "Vision-based approximate estimation of muscle activation patterns for tele-impedance," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5220–5227, 2023.

[29] K. H. Schulleri, L. Johannsen, Y. Michel, and D. Lee, "Sex differences in the association of postural control with indirect measures of body representations," *Scientific Reports*, vol. 12, no. 1, 2022.

[30] I. Tannert, K. H. Schulleri, Y. Michel, S. Villa, L. Johannsen, J. Hermsdorfer, and D. Lee, "Immediate effects of vibrotactile biofeedback instructions on human postural control," *43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 7426–7432, 2021.

[31] M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 119–125, 1981.

[32] J. Luh, M. Walker, and R. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Transactions on Automatic Control*, vol. 25, no. 3, pp. 468–474, 1980.

[33] B. Paden and R. Panja, "Globally asymptotically stable 'PD' controller for robot manipulators," *International Journal of Control*, vol. 47, no. 6, pp. 1697–1712, 1988.

[34] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.

[35] M. W. Spong, "Modeling and control of elastic joint robots," *Journal of Dynamic Syst., Measurement, and Control*, vol. 109, no. 4, pp. 310–318, 1987.

[36] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[37] M. H. Raibert and J. J. Craig, "Hybrid Position/Force Control of Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 1981.

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

[39] S. Chiaverini and L. Sciavicco, "The parallel approach to force/position control of robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 4, pp. 361–373, 1993.

[40] J. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," *19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 1980.

[41] K. Lee and M. Buss, "Force tracking impedance control with variable target stiffness," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 6751–6756, 2008.

[42] D. E. Whitney, "Quasi-static assembly of compliantly supported rigid parts," *Journal of Dynamic Systems, Measurement, and Control*, vol. 104, no. 1, pp. 65–77, 1982.

[43] A. D. Luca and W. J. Book, "Robots with flexible elements," *Springer Handbook of Robotics*, pp. 243–282, 2016.

[44] B. Vanderborght, A. Albu-Schaeffer, A. Bicchi, E. Burdet, D. Caldwell, R. Carloni, M. Catalano, O. Eiberger, W. Friedl, G. Ganesh, M. Garabini, M. Grebenstein, G. Grioli, S. Haddadin, H. Hoppner, A. Jafari, M. Laffranchi, D. Lefeber, F. Petit, S. Stramigioli, N. Tsagarakis, M. V. Damme, R. V. Ham, L. Visser, and S. Wolf, "Variable impedance actuators: A review," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1601–1614, 2013.

[45] C. Ott, R. Mukherjee, and Y. Nakamura, "A hybrid system framework for unified impedance and admittance control," *Journal of Intelligent And Robotic Systems*, vol. 78, no. 3-4, pp. 359–375, 2014.

[46] P. M. Rack and D. Westbury, "The short range stiffness of active mammalian muscle and its effect on mechanical properties," *The Journal of physiology*, vol. 240, no. 2, pp. 331–350, 1974.

[47] T. Nichols and J. Houk, "Improvement in linearity and regulation of stiffness that results from actions of stretch reflex," *journal of Neurophysiology*, vol. 39, no. 1, pp. 119–142, 1976.

[48] F. A. Mussa-Ivaldi, N. Hogan, and E. Bizzi, "Neural, mechanical, and geometric factors subserving arm posture in humans," *Journal of neuroscience*, vol. 5, no. 10, pp. 2732–2743, 1985.

[49] N. Hogan, "The mechanics of multi-joint posture and movement control," *Biological cybernetics*, vol. 52, no. 5, pp. 315–331, 1985.

[50] H. Gomi and R. Osu, "Task-dependent viscoelasticity of human multijoint arm and its spatial characteristics for interaction with environments," *Journal of neuroscience*, vol. 18, no. 21, pp. 8965–8978, 1998.

[51] A. Jacks, A. Prochazka, and P. Trend, "Instability in human forearm movements studied with feed-back-controlled electrical stimulation of muscles." *The Journal of physiology*, vol. 402, no. 1, pp. 443–461, 1988.

[52] J. R. Flanagan, E. Nakano, H. Imamizu, R. Osu, T. Yoshioka, and M. Kawato, "Composition and decomposition of internal models in motor learning under altered kinematic and dynamic environments," *The Journal of Neuroscience*, vol. 19, no. 20, 1999.

[53] M. Kawato, "Internal models for motor control and trajectory planning," *Current opinion in neurobiology*, vol. 9, no. 6, pp. 718–727, 1999.

[54] R. Shadmehr and F. A. Mussa-Ivaldi, "Adaptive representation of dynamics during learning of a motor task," *Journal of neuroscience*, vol. 14, no. 5, pp. 3208–3224, 1994.

[55] D. Rancourt and N. Hogan, "Stability in force-production tasks," *Journal of motor behavior*, vol. 33, no. 2, pp. 193–204, 2001.

[56] E. Burdet, K. P. Tee, I. Mareels, T. E. Milner, C.-M. Chew, D. W. Franklin, R. Osu, and M. Kawato, "Stability and motor adaptation in human arm movements," *Biological cybernetics*, vol. 94, pp. 20–32, 2006.

[57] K. P. Tee, D. W. Franklin, M. Kawato, T. E. Milner, and E. Burdet, "Concurrent adaptation of force and impedance in the redundant muscle system," *Biological cybernetics*, vol. 102, pp. 31–44, 2010.

[58] D. W. Franklin, R. Osu, E. Burdet, M. Kawato, and T. E. Milner, "Adaptation to stable and unstable dynamics achieved by combined impedance control and inverse dynamics model," *Journal of neurophysiology*, vol. 90, no. 5, pp. 3270–3282, 2003.

[59] G. Ganesh, A. Albu-Schäffer, M. Haruno, M. Kawato, and E. Burdet, "Biomimetic motor behavior for simultaneous adaptation of force, impedance and trajectory in interaction tasks," *IEEE International Conference on Robotics and Automation*, pp. 2705–2711, 2010.

[60] E. Burdet, "Algorithms of human motor control and their implementation in robotics," Ph.D. dissertation, ETH Zurich, 1996.

[61] J. Blaya and H. Herr, "Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, no. 1, pp. 24–31, 2004.

[62] X. Li, Y. Pan, G. Chen, and H. Yu, "Multi-modal control scheme for rehabilitation robotic exoskeletons," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 759–777, 2017.

[63] R. Wu and A. Billard, "Learning from demonstration and interactive control of variable-impedance to cut soft tissues," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 2740–2751, 2022.

[64] 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.

[65] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2017.

[66] N. Mansfeld, M. Keppler, and S. Haddadin, "Speed gain in elastic joint robots: An energy conversion-based approach," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4600–4607, 2021.

[67] S. Haddadin, T. Laue, U. Frese, S. Wolf, A. Albu-Schäffer, and G. Hirzinger, "Kick it with elasticity: Safety and performance in human–robot soccer," *Robotics and Autonomous Systems*, vol. 57, no. 8, pp. 761–775, 2009.

[68] G. Pratt and M. Williamson, "Series elastic actuators," *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots.*

[69] S. Wolf and G. Hirzinger, "A new variable stiffness design: Matching requirements of the next robot generation," *IEEE International Conference on Robotics and Automation*, pp. 1741–1746, 2008.

[70] S. Wolf, G. Grioli, O. Eiberger, W. Friedl, M. Grebenstein, H. Hoppner, E. Burdet, D. G. Caldwell, R. Carloni, M. G. Catalano, D. Lefeber, S. Stramigioli, N. Tsagarakis, M. V. Damme, R. V. Ham, B. Vanderborght, L. C. Visser, A. Bicchi, and A. Albu-Schäffer, "Variable stiffness actuators: Review on design

and components," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 5, pp. 2418–2430, 2016.

[71] M. Grebenstein, A. Albu-Schäffer, T. Bahls, M. Chalon, O. Eiberger, W. Friedl, R. Gruber, S. Haddadin, U. Hagn, R. Haslinger, H. Hoppner, S. Jorg, M. Nickl, A. Nothhelfer, F. Petit, J. Reill, N. Seitz, T. Wimbock, S. Wolf, T. Wusthoff, and G. Hirzinger, "The DLR hand arm system," *IEEE International Conference on Robotics and Automation*, 2011.

[72] R. Ikeura and H. Inooka, "Variable impedance control of a robot for cooperation with a human," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 3097–3102, 1995.

[73] R. Ikeura, T. Moriguchi, and K. Mizutani, "Optimal variable impedance control for a robot and its application to lifting an object with a human," *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication*, pp. 500–505, 2002.

[74] T. Tsumugiwa, R. Yokogawa, and K. Hara, "Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 644–650, 2002.

[75] V. Duchaine and C. Gosselin, "Safe, stable and intuitive control for physical human-robot interaction," *IEEE International Conference on Robotics and Automation*, pp. 3383–3388, 2009.

[76] J. R. Medina, H. Borner, S. Endo, and S. Hirche, "Impedance-based gaussian processes for modeling human motor behavior in physical and non-physical interaction," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 9, pp. 2499–2511, 2019.

[77] S. Calinon and D. Lee, "Learning control," P. Vadakkepat and A. Goswami, Eds., 2019, pp. 1261–1312.

[78] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.

[79] 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.

[80] L. Rozo, D. Bruno, S. Calinon, and D. G. Caldwell, "Learning optimal controllers in human-robot cooperative transportation tasks with position and force constraints," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1024–1030, 2015.

[81] S. Calinon, "Gaussians on riemannian manifolds: Applications for robot learning and adaptive control," *IEEE Robotics And Automation Magazine*, vol. 27, no. 2, pp. 33–45, 2020.

[82] F. J. Abu-Dakka and V. Kyrki, "Geometry-aware dynamic movement primitives," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4421–4426, 2020.

[83] S. Schaal, "Dynamic movement primitives -a framework for motor control in humans and humanoid robotics," *Adaptive Motion of Animals and Machines*, pp. 261–280, 2006.

[84] K. Kronander and A. Billard, "Online learning of varying stiffness through physical human-robot interaction," *IEEE International Conference on Robotics and Automation*, pp. 1842–1849, 2012.

[85] C. Yang, C. Zeng, Y. Cong, N. Wang, and M. Wang, "A learning framework of adaptive manipulative skills from human to robot," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1153–1161, 2019.

[86] C. Yang, C. Zeng, C. Fang, W. He, and Z. Li, "A DMPs-based framework for robot learning and generalization of humanlike variable impedance skills," *IEEE/ASME Trans. on Mechatronics*, vol. 23, no. 3, pp. 1193–1203, 2018.

[87] C. Yang, C. Zeng, P. Liang, Z. Li, R. Li, and C. Su, "Interface design of a physical human-robot interaction system for human impedance adaptive skill transfer," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 329–340, 2018.

[88] L. Peternel, L. Rozo, D. Caldwell, and A. Ajoudani, "A method for derivation of robot task-frame control authority from repeated sensory observations," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 719–726, 2017.

[89] R. Wu, H. Zhang, and J. Zhao, "Robot variable impedance skill transfer and learning framework based on a simplified human arm impedance model," *IEEE Access*, vol. 8, pp. 225 627–225 638, 2020.

[90] Y. Wu, F. Zhao, T. Tao, and A. Ajoudani, "A framework for autonomous impedance regulation of robots based on imitation learning and optimal control," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 127–134, 2021.

[91] S. Calinon, I. Sardellitti, and D. G. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 249–254, 2010.

[92] M. Saveriano and D. Lee, "Learning motion and impedance behaviors from human demonstrations," *11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 368–373, 2014.

[93] D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Autonomous Robots*, vol. 31, pp. 115–131, 2011.

[94] T. Kastritsi, F. Dimeas, and Z. Doulgeri, "Progressive automation with dmp synchronization and variable stiffness control," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3789–3796, 2018.

[95] F. Dimeas and Z. Doulgeri, "Progressive automation of periodic tasks on planar surfaces of unknown pose with hybrid force/position control," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5246–5252, 2020.

[96] F. Ferraguti, C. Secchi, and C. Fantuzzi, "A tank-based approach to impedance control with variable stiffness," *IEEE International Conference on Robotics and Automation*, pp. 4948–4953, 2013.

[97] P. Kormushev, S. Calinon, and D. G. Caldwell, "Robot motor skill coordination with EM-based reinforcement learning," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3232–3237, 2010.

[98] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 918–930, 2011.

[99] Y. Li, G. Ganesh, N. Jarrassé, S. Haddadin, A. Albu-Schaeffer, and E. Burdet, "Force, impedance, and trajectory learning for contact tooling and haptic identification," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1170–1182, 2018.

[100] E. Gribovskaya, A. Kheddar, and A. Billard, "Motion learning and adaptive impedance for robot control during physical interaction with humans," *IEEE International Conference on Robotics and Automation*, pp. 4326–4332, 2011.

[101] L. Johannsmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," *International Conference on Robotics and Automation (ICRA)*, pp. 5844–5850, 2019.

[102] M. Song and H. Wang, "Highly efficient incremental estimation of gaussian mixture models for online data stream clustering," *SPIE Proceedings*, 2005.

[103] M. J. Mathew, S. Sidhik, M. Sridharan, M. Azad, A. Hayashi, and J. Wyatt, "Online learning of feed-forward models for task-space variable impedance control," *IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pp. 222–229, 2019.

[104] B. Kim, J. Park, S. Park, and S. Kang, "Impedance learning for robotic contact tasks using natural actor-critic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 2, pp. 433–443, 2010.

[105] J. Peters and S. Schaal, "Natural actor-critic," *Neurocomputing*, vol. 71, no. 7-9, pp. 1180–1190, 2008.

[106] J. Buchli, E. Theodorou, F. Stulp, and S. Schaal, "Variable impedance control - a reinforcement learning approach," *Robotics: Science and Systems*, pp. 153–160, 2010.

[107] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *The Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.

[108] F. Winter, M. Saveriano, and D. Lee, "The role of coupling terms in variable impedance policies learning," *9th International Workshop on Human-Friendly Robotics (HFR)*, 2016.

[109] J. Rey, K. Kronander, F. Farshidian, J. Buchli, and A. Billard, "Learning motions from demonstrations and rewards with time-invariant dynamical systems based policies," *Autonomous Robots*, vol. 42, 2018.

[110] S. A. Khader, H. Yin, P. Falco, and D. Kragic, "Stability-guaranteed reinforcement learning for contact-rich manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 1–8, 2021.

[111] M. Khansari, K. Kronander, and A. Billard, "Modeling robot discrete movements with state-varying stiffness and damping: A framework for integrated motion generation and impedance control," *Robotics: Science and Systems*, 2014.

[112] A. Van der Schaft, *L2-gain and passivity techniques in nonlinear control*. Springer, 2000.

[113] F. Ferraguti, N. Preda, A. Manurung, M. Bonfé, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini, and C. Secchi, "An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1073–1088, 2015.

[114] K. Kronander and A. Billard, "Stability considerations for variable impedance control," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1298–1305, 2016.

[115] T. Sun, L. Peng, L. Cheng, Z.-G. Hou, and Y. Pan, "Stability-guaranteed variable impedance control of robots based on approximate dynamic inversion," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 7, pp. 4193–4200, 2021.

[116] E. Spyrakos-Papastavridis, P. R. N. Childs, and J. S. Dai, "Passivity preservation for variable impedance control of compliant robots," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2342–2353, 2020.

[117] F. Ficuciello, L. Villani, and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive human-robot physical interaction," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 850–863, 2015.

[118] D. S. Walker, J. K. Salisbury, and G. Niemeyer, "Demonstrating the benefits of variable impedance to telerobotic task execution," *IEEE International Conference on Robotics and Automation*, pp. 1348–1353, 2011.

[119] A. Takagi, G. Xiong, H. Kambara, and Y. Koike, "Endpoint stiffness magnitude increases linearly with a stronger power grasp," *Scientific Reports*, vol. 10, no. 1, 2020.

[120] D. S. Walker, R. P. Wilson, and G. Niemeyer, "User-controlled variable impedance teleoperation," *IEEE International Conference on Robotics and Automation*, pp. 5352–5357, 2010.

[121] M. Laghi, A. Ajoudani, M. G. Catalano, and A. Bicchi, "Unifying bilateral teleoperation and tele-impedance for enhanced user experience," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 514–539, 2020.

[122] B. Hannaford and J.-H. Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, 2002.

[123] Jee-Hwan Ryu, Dong-Soo Kwon, and B. Hannaford, "Stable teleoperation with time-domain passivity control," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 365–373, 2004.

[124] G. Gourmelen, B. Navarro, A. Cherubini, and G. Ganesh, "Human guided trajectory and impedance adaptation for tele-operated physical assistance," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9276–9282, 2021.

[125] L. Peternel, T. Petrič, and J. Babič, "Robotic assembly solution by human-in-the-loop teaching method based on real-time stiffness modulation," *Autonomous Robots*, vol. 42, pp. 1–17, 2018.

[126] Y.-C. Huang, D. A. Abbink, and L. Peternel, "A semi-autonomous tele-impedance method based on vision and voice interfaces," *International Conference on Advanced Robotics (ICAR)*, pp. 180–186, 2021.

[127] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.

[128] D. Kulic, D. Lee, C. Ott, and Y. Nakamura, "Incremental learning of full body motion primitives for humanoid robots," *IEEE-RAS International Conference on Humanoid Robots*, pp. 326–332, 2008.

[129] D. Lee, C. Ott, and Y. Nakamura, "Mimetic communication model with compliant physical contact in human—humanoid interaction," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1684–1704, 2010.

[130] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 408–423, 2015.

[131] K. Kronander and A. Billard, "Passive interaction control with dynamical systems," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 106–113, 2016.

[132] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for safe robots: Measurements, analysis and new insights," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, 2009.

[133] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. How, and G. Fiore, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[134] M. Mason, "The mechanics of manipulation," *IEEE International Conference on Robotics and Automation*, pp. 544–548, 1985.

[135] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, 1999.

[136] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[137] S. Khansari-Zadeh and O. Khatib, "Learning potential functions from human demonstrations with encapsulated dynamic and compliant behaviors," *Autonomous Robots*, vol. 41, pp. 45–69, 2017.

[138] G. Franzese, A. Mészéros, L. Peternel, and J. Kober, "Ilosa: Interactive learning of stiffness and attractors," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7778–7785, 2021.

[139] P. Li and R. Horowitz, "Passive velocity field control of mechanical manipulators," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, pp. 751–763, 1999.

[140] V. Duindam and S. Stramigioli, "Port-based asymptotic curve tracking for mechanical systems," *European Journal of Control*, vol. 10, no. 5, pp. 411–420, 2004.

[141] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx, *Modeling and control of complex physical systems: the port-Hamiltonian approach*. Springer Science & Business Media, 2009.

[142] H. M. Paynter, "Analysis and design of engineering systems," *MIT press*, 1961.

[143] R. Ortega, A. Van Der Schaft, I. Mareels, and B. Maschke, "Putting energy back in control," *IEEE Control Systems*, vol. 21, no. 2, pp. 18–33, 2001.

[144] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and control of legged robots," *Springer Handbook of Robotics*, pp. 1203–1234, 2016.

[145] B. Henze, M. A. Roa, and C. Ott, "Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1522–1543, 2016.

[146] R. Ortega, I. Mareels, A. van der Schaft, and B. Maschke, "Energy shaping revisited," *International Conference on Control Applications*, pp. 121–126, 2000.

[147] R. Ortega, A. van der Schaft, B. Maschke, and G. Escobar, "Interconnection and damping assignment passivity-based control of port-controlled hamiltonian systems," *Automatica*, vol. 38, no. 4, pp. 585–596, 2002.

[148] J. Willems, "Paradigms and puzzles in the theory of dynamical systems," *IEEE Transactions on Automatic Control*, vol. 36, no. 3, pp. 259–294, 1991.

[149] S. Stramigioli, "Energy-aware robotics," *Mathematical Control Theory I*, pp. 37–50, 2015.

[150] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, "Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 741–756, 2011.

[151] M. Selvaggio, P. Robuffo Giordano, F. Ficuciello, and B. Siciliano, "Passive task-prioritized shared-control teleoperation with haptic guidance," *International Conference on Robotics and Automation*, pp. 430–436, 2019.

[152] C. Schindlbeck and S. Haddadin, "Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks," *IEEE International Conference on Robotics and Automation*, pp. 440–447, 2015.

[153] R. Rashad, J. B. C. Engelen, and S. Stramigioli, "Energy tank-based wrench/impedance control of a fully-actuated hexarotor: A geometric port-hamiltonian approach," *International Conference on Robotics and Automation*, pp. 6418–6424, 2019.

[154] A. Dietrich, C. Ott, and S. Stramigioli, "Passivation of projection-based null space compliance control via energy tanks," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. pp. 184–191, 2016.

[155] *ISO TC184 SC2, ISO/TS 15066 Robots and robotic devices-Safety requirements for industrial robots-Collaborative operation*, 2013.

[156] J. Wood, "Dynamic response of human cranial bone," *Journal of Biomechanics*, vol. 4, no. 1, pp. 1–12, 1971.

[157] N. Yoganandan, F. Pintar, D. Maiman, J. Cusick, A. Sances, and P. Walsh, "Human head-neck biomechanics under axial tension," *Medical Engineering Physics*, vol. 18, no. 4, pp. 289–294, 1996.

[158] G. Raiola, C. A. Cardenas, T. S. Tadele, T. de Vries, and S. Stramigioli, "Development of a safety- and energy-aware impedance controller for collaborative robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1237–1244, 2018.

[159] M. Geravand, E. Shahriari, A. De Luca, and A. Peer, "Port-based modeling of human-robot collaboration towards safety-enhancing energy shaping control," *IEEE International Conference on Robotics and Automation*, pp. 3075–3082, 2016.

[160] M. Laffranchi, N. G. Tsagarakis, and D. G. Caldwell, "Safe human robot interaction via energy regulation control," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 35–41, 2009.

[161] E. Shahriari, D. Zardykhan, A. Koenig, E. Jensen, and S. Haddadin, "Energy-based adaptive control and learning for patient-aware rehabilitation," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5671–5678, 2019.

[162] E. Shahriari, A. Kramberger, A. Gams, A. Ude, and S. Haddadin, "Adapting to contacts: Energy tanks and task energy for passivity-based dynamic movement primitives," *IEEE-RAS International Conference on Humanoid Robotics (Humanoids)*, pp. 6023–6028, 2017.

[163] A. Kramberger, E. Shahriari, A. Gams, B. Nemec, A. Ude, and S. Haddadin, "Passivity based iterative learning of admittance-coupled dynamic movement primitives for interaction with changing environments," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[164] E. Shahriari, L. Johannsmeier, and S. Haddadin, "Valve-based virtual energy tanks: A framework to simultaneously passify controls and embed control objectives," *Annual American Control Conference (ACC)*, pp. 3634–3641, 2018.

[165] E. Shahriari, L. Johannsmeier, E. Jensen, and S. Haddadin, "Power flow regulation, adaptation, and learning for intrinsically robust virtual energy tanks," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 211–218, 2020.

[166] R. C. Goertz, "A force-reflecting positional servomechanism," *Nucleonics*, vol. 10, no. 11, pp. 43–45, 1952.

[167] R. Ambrose, H. Aldridge, R. Askew, R. Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark, "Robonaut: NASA's space humanoid," *IEEE Intelligent Systems*, vol. 15, no. 4, pp. 57–63, 2000.

[168] J. Artigas, R. Balachandran, C. Riecke, M. Stelzer, B. Weber, J.-H. Ryu, and A. Albu-Schaeffer, "KONTUR-2: Force-feedback teleoperation from the international space station," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[169] L. Whitcomb, "Underwater robotics: out of the research laboratory and into the field," *International Conference on Robotics and Automation*, pp. 709–716, 2000.

[170] Q. Lin and C. Kuo, "Virtual tele-operation of underwater robots," *International Conference on Robotics and Automation*, pp. 1022–1027, 1997.

[171] M. Talha, A. M. Ghalamzan E., C. Takahashi, J. Kuo, W. Ingamells, and R. Stolkin, "Towards robotic decommissioning of legacy nuclear plant: Results of human-factors experiments with tele-robotic manipulation, and a discussion of challenges and approaches for decommissioning," *International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 166–173, 2016.

[172] R. Bogue, "Robots in the nuclear industry: a review of technologies and applications," *Industrial Robot: An International Journal*, vol. 38, no. 2, pp. 113–118, 2011.

[173] G. Ballantyne, "Robotic surgery, telerobotic surgery, telepresence, and telementoring," *Surgical Endoscopy*, vol. 16, no. 10, pp. 1389–1402, 2002.

[174] R. V. Patel, S. F. Atashzar, and M. Tavakoli, "Haptic feedback and force-based teleoperation in surgical robotics," *Proceedings of the IEEE*, vol. 110, no. 7, pp. 1012–1027, 2022.

[175] C. Preusche, T. Ortmaier, and G. Hirzinger, "Teleoperation concepts in minimal invasive surgery," *Control Engineering Practice*, vol. 10, no. 11, pp. 1245–1250, 2002.

[176] O. Khatib, X. Yeh, G. Brantner, B. Soe, B. Kim, S. Ganguly, H. Stuart, S. Wang, M. Cutkosky, A. Edsinger, *et al.*, "Ocean one: A robotic avatar for oceanic discovery," *IEEE Robotics and Automation Magazine*, vol. 23, no. 4, pp. 20–29, 2016.

[177] J. Marescaux, J. Leroy, F. Rubino, M. Smith, M. Vix, M. Simone, and D. Mutter, "Transcontinental robot-assisted remote telesurgery: Feasibility and potential applications," *Annals of Surgery*, vol. 235, no. 4, pp. 487–492, 2002.

[178] S. O'Sullivan, N. Nevejans, C. Allen, A. Blyth, S. Leonard, U. Pagallo, K. Holzinger, A. Holzinger, M. I. Sajid, and H. Ashrafian, "Legal, regulatory, and ethical frameworks for development of standards in artificial intelligence (AI) and autonomous robotic surgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 15, no. 1, p. e1968, 2019.

[179] L. M. Parsons, "Inability to reason about an object's orientation using an axis and angle of rotation." *Journal of experimental psychology: Human perception and performance*, vol. 21, no. 6, p. 1259, 1995.

[180] D. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 624–637, 1993.

[181] T. B. Sheridan, "Human–robot interaction: status and challenges," *Human factors*, vol. 58, no. 4, pp. 525–532, 2016.

[182] G. Niemeyer, C. Preusche, S. Stramigioli, and D. Lee, "Telerobotics," *Springer handbook of robotics*, pp. 1085–1108, 2016.

[183] W. R. Ferrell and T. B. Sheridan, "Supervisory control of remote manipulation," *IEEE Spectrum*, vol. 4, no. 10, pp. 81–88, 1967.

[184] P. Schmaus, D. Leidner, T. Krüger, A. Schiele, B. Pleintinger, R. Bayer, and N. Y. Lii, "Preliminary insights from the meteron supvis justin space-robotics experiment," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3836–3843, 2018.

[185] C. Borst, T. Wimbock, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. R. Giordano, R. Konietschke, W. Sepp, S. Fuchs, C. Rink, A. Albu-Schaffer, and G. Hirzinger, "Rollin' justin - mobile platform with variable base," *2009 IEEE International Conference on Robotics and Automation*, 2009.

[186] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano, "Autonomy in physical human-robot interaction: A brief survey," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7989–7996, 2021.

[187] D. P. Losey, C. G. Mcdonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human-robot interaction," *Applied Mechanics Reviews*, vol. 70, pp. 010 804–010 804, 2018.

[188] Y. Li, A. Takagi, and K. P. Tee, "Editorial: Shared control for tele-operation systems," *Frontiers in Robotics and AI*, vol. 9, 2022.

[189] T. Ortmaier, M. Groger, D. Boehm, V. Falk, and G. Hirzinger, "Motion estimation in beating heart surgery," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 10, pp. 1729–1740, 2005.

[190] F. Abi-Farrajl, B. Henze, A. Werner, M. Panzirsch, C. Ott, and M. A. Roa, "Humanoid teleoperation using task-relevant haptic feedback," *International Conference on Intelligent Robots and Systems (IROS)*, pp. 5010–5017, 2018.

[191] F. Abi-Farraj, N. Pedemonte, and P. Robuffo Giordano, "A visual-based shared control architecture for remote telemanipulation," *International Conference on Intelligent Robots and Systems (IROS)*, pp. 4266–4273, 2016.

[192] I. Havoutis and S. Calinon, "Learning from demonstration for semi-autonomous teleoperation," *Autonomous Robots*, vol. 43, pp. 713–726, 2019.

[193] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: smoothly shifting control authority?" *Cognition, Technology and Work*, vol. 14, no. 1, pp. 19–28, 2012.

[194] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[195] L. B. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," *Proc. IEEE Virtual Reality Annual Int. Symposium*, pp. 76–82, 1993.

[196] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," *Robotics Research: Results of the 12th International Symposium ISRR*, pp. 49–64, 2007.

[197] D. Zhang, Z. Wu, J. Chen, R. Zhu, A. Munawar, B. Xiao, Y. Guan, H. Su, W. Hong, Y. Guo, G. S. Fischer, B. Lo, and G.-Z. Yang, "Human-robot shared control for surgical robot based on context-aware sim-to-real adaptation," *International Conference on Robotics and Automation*, pp. 7694–7700, 2022.

[198] G. Quere, S. Bustamante, A. Hagengruber, J. Vogel, F. Steinmetz, and F. Stulp, "Learning and interactive design of shared control templates," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1887–1894, 2021.

[199] G. Quere, A. Hagengruber, M. Iskandar, S. Bustamante, D. Leidner, F. Stulp, and J. Vogel, "Shared control templates for assistive robotics," *IEEE International Conference on Robotics and Automation*, pp. 1956–1962, 2020.

[200] J. Vogel, A. Hagengruber, M. Iskandar, G. Quere, U. Leipscher, S. Bustamante, A. Dietrich, H. Höppner, D. Leidner, and A. Albu-Schäffer, "Edan: An emg-controlled daily assistant to help people with physical disabilities," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4183–4190, 2020.

[201] I. Havoutis and S. Calinon, "Learning assistive teleoperation behaviors from demonstration," *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 258–263, 2016.

[202] I. Havoutis, A. Tanwani, and S. Calinon, "Online incremental learning of manipulation tasks for semi-autonomous teleoperation," *International Conference on Intelligent Robots and Systems, Workshop on Closed Loop Grasping and Manipulation: Challenges and Progress*, 2016.

[203] M. J. Zeestraten, I. Havoutis, and S. Calinon, "Programming by demonstration for shared control with an application in teleoperation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1848–1855, 2018.

[204] G. Raiola, S. Sanchez Restrepo, P. Chevalier, P. Rodriguez, X. Lamy, S. Tliba, and F. Stulp, "Co-manipulation with a library of virtual guiding fixtures," *Autonomous Robots*, vol. 42, 2018.

[205] G. Raiola, X. Lamy, and F. Stulp, "Co-manipulation with multiple probabilistic virtual guides," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1037–1051, 2015.

[206] M. Jamsek, T. Kunavar, U. Bobek, E. Rueckert, and J. Babic, "Predictive exoskeleton control for arm-motion augmentation based on probabilistic movement primitives combined with a flow controller," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4417–4424, 2021.

[207] A. Martínez, B. Lawson, C. Durrough, and M. Goldfarb, "A velocity-field-based controller for assisting leg movement during walking with a bilateral hip and knee lower limb exoskeleton," *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 307–316, 2019.

[208] F. Abi-Farraj, T. Osa, N. P. J. Peters, G. Neumann, and P. R. Giordano, "A learning-based shared control architecture for interactive task execution," *IEEE international conference on robotics and automation (ICRA)*, pp. 329–335, 2017.

[209] A. Pervez, H. Latifee, J.-H. Ryu, and D. Lee, "Motion encoding with asynchronous trajectories of repetitive teleoperation tasks and its extension to human-agent shared teleoperation," *Autonomous Robots*, vol. 43, no. 8, pp. 2055–2069, 2019.

[210] M. Cheng, R. Li, R. Ding, and S. Luo, "Dynamic guidance virtual fixture for hydraulic manipulator via learning from demonstration," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 237, no. 4, pp. 952–962, 2023.

[211] H. Ishida, M. M. Marinho, K. Harada, J. Gao, and M. Mitsuishi, "Virtual-fixtures for robotic-assisted bi-manual cutting using vector-field inequalities," *IEEE/SICE International Symposium on System Integration*, pp. 395–400, 2020.

[212] 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, pp. 524–531, 2019.

[213] R. Rahal, F. Abi-Farraj, P. R. Giordano, and C. Pacchierotti, "Haptic shared-control methods for robotic cutting under nonholonomic constraints," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8151–8157, 2019.

[214] Z. Chen, A. Malpani, P. Chalasani, A. Deguet, S. S. Vedula, P. Kazanzides, and R. H. Taylor, "Virtual fixture assistance for needle passing and knot tying," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2343–2350, 2016.

[215] R. Moccia, M. Selvaggio, L. Villani, B. Siciliano, and F. Ficuciello, "Vision-based virtual fixtures generation for robotic-assisted polyp dissection procedures," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7934–7939, 2019.

[216] X. Duan, H. Tian, C. Li, Z. Han, T. Cui, Q. Shi, H. Wen, and J. Wang, "Virtual-fixture based drilling control for robot-assisted craniotomy: Learning from demonstration," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2327–2334, 2021.

[217] C. J. Perez-del Pulgar, J. Smisek, V. F. Munoz, and A. Schiele, "Using learning from demonstration to generate real-time guidance for haptic shared control," *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3205–3210, 2016.

[218] M. Hagenow, E. Senft, R. Radwin, M. Gleicher, B. Mutlu, and M. Zinn, "Corrective shared autonomy for addressing task variability," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3720–3727, 2021.

[219] H. K. Khalil, *Nonlinear systems*. Pearson, 2001.

[220] J.-J. E. Slotine, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991.

[221] C. Secchi, S. Stramigioli, and C. Fantuzzi, *Control of Interactive Robotic Interfaces: A Port-Hamiltonian Approach*. Springer, 2007.

[222] F. Califano, R. Rashad, C. Secchi, and S. Stramigioli, "On the use of energy tanks for robotic systems," *Human-Friendly Robotics 2022*, pp. 174–188, 2023.

[223] E. Colgate and N. Hogan, "An analysis of contact instability in terms of passive physical equivalents," *International Conference on Robotics and Automation*, pp. 404–409 vol.1, 1989.

[224] C. Ott, A. Kugi, and Yoshihiko Nakamura, "Resolving the problem of non-integrability of nullspace velocities for compliance control of redundant manipulators by using semi-definite lyapunov functions," *IEEE International Conference on Robotics and Automation*, pp. 1999–2004, 2008.

[225] C. Ott, A. Dietrich, and A. Albu-Schäffer, "Prioritized multi-task compliance control of redundant manipulators," *Automatica*, vol. 53, pp. 416 – 423, 2015.

[226] J. Park, W. Chung, and Y. Youm, "On dynamical decoupling of kinematically redundant manipulators," *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289)*, vol. 3, pp. 1495–1500, 1999.

[227] D. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 354–369, 2010.

[228] J. A. Newman, N. Shewchenko, and E. Welbourne, "A proposed new biomechanical head injury assessment function - the maximum power index." *Stapp car crash journal*, vol. 44, pp. 215–47, 2000.

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

[230] S. Stramigioli, *Modeling and IPC Control of Interactive Mechanical Systems: A Coordinate-free Approach*, ser. Lecture Notes in Control and Information Sciences, 2001.

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

[232] G. B. Dantzig and M. N. Thapa, *Linear programming 1: introduction*. Springer New York, NY, 2006.

[233] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *The International Journal of Robotics Research*, vol. 14, no. 1, pp. 19–36, 1995.

[234] N. Mansfeld, Y. Michel, T. Bruckmann, and S. Haddadin, "Improving the performance of auxiliary null space tasks via time scaling-based relaxation of the primary task," *International Conference on Robotics and Automation (ICRA)*, pp. 9342–9348, 2019.

[235] G. Garofalo and C. Ott, "Passive energy-based control via energy tanks and release valve for limit cycle and compliance control," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 73–78, 2018.

[236] C. Ott, J. Artigas, and C. Preusche, "Subspace-oriented energy distribution for the time domain passivity approach," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 665–671, 2011.

[237] K. J. Kuchenbecker and G. Niemeyer, "Modeling induced master motion in force-reflecting teleoperation," *IEEE International Conference on Robotics and Automation*, pp. 348–353, 2005.

[238] K. J. Kuchenbecker and G. Niemeyer, "Induced Master Motion in Force-Reflecting Teleoperation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 4, pp. 800–810, 2006.

[239] D. Heck, "Delayed bilateral teleoperation : a direct force-reflecting control approach," Ph.D. dissertation, Technische Universiteit Eindhoven, 2015.

[240] C. Pacchierotti, A. Tirmizi, G. Bianchini, and D. Prattichizzo, "Enhancing the performance of passive teleoperation systems via cutaneous feedback," *IEEE Transactions on Haptics*, vol. 8, no. 4, pp. 397–409, 2015.

[241] L. Zhang, M. M. Diraneyya, J. Ryu, C. T. Haas, and E. M. Abdel-Rahman, "Jerk as an indicator of physical exertion and fatigue," *Automation in Construction*, vol. 104, pp. 120–128, 2019.

[242] 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, 2010.

[243] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[244] M. Saveriano and D. Lee, "Distance based dynamical system modulation for reactive avoidance of moving obstacles," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5618–5623, 2014.

[245] S. S. M. Salehian, M. Khoramshahi, and A. Billard, "A dynamical system approach for softly catching a flying object: Theory and experiment," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 462–471, 2016.

[246] W. Amanhoud, M. Khoramshahi, M. Bonnesoeur, and A. Billard, "Force adaptation in contact tasks with dynamical systems," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6841–6847, 2020.

[247] M. Saveriano and D. Lee, "Learning barrier functions for constrained motion planning with dynamical systems," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 112–119, 2019.

[248] H. Khurana, M. Bombile, and A. Billard, "Learning to hit: A statistical dynamical system based approach," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[249] S. M. Khansari-Zadeh and A. Billard, "Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[250] K. Kronander, M. Khansari, and A. Billard, "Incremental motion learning with locally modulated dynamical systems," *Robotics and Autonomous Systems*, vol. 70, pp. 52–62, 2015.

[251] W. Xiao, A. Lederer, and S. Hirche, "Learning stable nonparametric dynamical systems with gaussian process regression," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1194–1199, 2020.

[252] N. B. Figueroa Fernandez and A. Billard, "Modeling compositions of impedance-based primitives via dynamical systems." *In Proceedings of the Workshop on Cognitive Whole-Body Control for Compliant Robot Manipulation (COWB-COMP)*, 2018.

[253] N. Figueroa and A. Billard, "Locally active globally stable dynamical systems: Theory, learning, and experiments," *The International Journal of Robotics Research*, vol. 41, no. 3, pp. 312–347, 2022.

[254] M. Khoramshahi and A. Billard, "A dynamical system approach for detection and reaction to human guidance in physical human-robot interaction," *Autonomous Robots*, vol. 44, no. 8, pp. 1411–1429, 2020.

[255] ——, "A dynamical system approach to task-adaptation in physical human–robot interaction," *Autonomous Robots*, vol. 43, no. 4, pp. 927–946, 2019.

[256] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3356–3363, 2008.

[257] J. Buchli, E. Theodorou, F. Stulp, and S. Schaal, "Variable impedance control a reinforcement learning approach," *Robotics: Science and Systems VI*, pp. 153–160, 2011.

[258] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," 2007, pp. 49–64.

[259] C. J. Wu and M. S. Hamada, *Experiments: planning, analysis, and optimization*. John Wiley & Sons, 2011.

[260] M. Saveriano, "An energy-based approach to ensure the stability of learned dynamical systems," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4407–4413, 2020.

[261] C. An, C. Atkeson, J. Griffiths, and J. Hollerbach, "Experimental evaluation of feedforward and computed torque control," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 368–373, 1989.

[262] S. M. Khansari-Zadeh, "The LASA handwriting dataset." [Online]. Available: http://bitbucket.org/khansari/lasahandwritingdataset

[263] N. Figueroa and A. Billard, "A physically-consistent bayesian non-parametric mixture model for dynamical system learning," pp. 927–946, 2018.

[264] A. Ude, "Filtering in a unit quaternion space for model-based object tracking," *Robotics and Autonomous Systems*, vol. 28, no. 2-3, pp. 163–172, 1999.

[265] A. Ude, B. Nemec, T. Petric, and J. Morimoto, "Orientation in cartesian space dynamic movement primitives," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2997–3004, 2014.

[266] W. Wang, M. Saveriano, and F. J. Abu-Dakka, "Learning deep robotic skills on riemannian manifolds," *IEEE Access*, vol. 10, pp. 114 143–114 152, 2022.

[267] M. Saveriano, F. J. Abu-Dakka, and V. Kyrki, "Learning stable robotic skills on riemannian manifolds," *Robotics and Autonomous Systems (accepted)*, 2022.

[268] M. J. Zeestraten, I. Havoutis, S. Calinon, and D. G. Caldwell, "Learning task-space synergies using riemannian geometry," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 73–78, 2017.

[269] A. Dietrich, C. Ott, and J. Park, "The hierarchical operational space formulation: Stability analysis for the regulation case," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1120–1127, 2018.

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

[271] J. Lachner, F. Allmendinger, E. Hobert, N. Hogan, and S. Stramigioli, "Energy budgets for coordinate invariant robot control in physical human–robot interaction," *The International Journal of Robotics Research*, vol. 40, no. 8-9, pp. 968–985, 2021.

[272] S. Haddadin, S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schäffer, "On making robots understand safety: Embedding injury knowledge into control," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1578–1602, 2012.