

Control and Learning of Compliant Manipulation Skills

THÈSE N° 6717 (2015)

PRÉSENTÉE LE 4 SEPTEMBRE 2015

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE D'ALGORITHMES ET SYSTÈMES D'APPRENTISSAGE
PROGRAMME DOCTORAL EN SYSTÈMES DE PRODUCTION ET ROBOTIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Klas Jonas Alfred KRONANDER

acceptée sur proposition du jury:

Prof. J. Paik, présidente du jury
Prof. A. Billard, directrice de thèse
Prof. E. Burdet, rapporteur
Prof. S. Stramigioli, rapporteur
Prof. J. Buchli, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

ABSTRACT

HUMANS demonstrate an impressive capability to manipulate fragile objects without damaging them, graciously controlling the force and position of hands or tools. Traditionally, robotics has favored position control over force control to produce fast, accurate and repeatable motion. For extending the applicability of robotic manipulators outside the strictly controlled environments of industrial work cells, position control is inadequate. Tasks that involve contact with objects whose positions are not known with perfect certainty require a controller that regulates the relationship between positional deviations and forces on the robot. This problem is formalized in the impedance control framework, which focuses the robot control problem on the interaction between the robot and its environment. By adjusting the impedance, the behavior of the robot can be adapted to the need of the task. However, it is often difficult to specify formally how the impedance should vary for best performance. Furthermore, fast it can be shown that careless variation of the impedance can lead to unstable regulation or tracking even in free motion.

In the first part of the thesis, the problem of how to define a varying impedance for a task is addressed. A haptic human-robot interface that allows a human supervisor to teach impedance variations by physically interacting with the robot during task execution is introduced. It is shown that the interface can be used to enhance the performance in several manipulation tasks. Then, the problem of stable control with varying impedance is addressed. Along with a theoretical discussion on this topic, a sufficient condition for stable varying stiffness and damping is provided.

In the second part of the thesis, we explore more complex manipulation scenarios via online generation of the robot trajectory. This is done along two axes 1) learning how to react to contact forces in insertion tasks which are crucial for assembly operations and 2) autonomous Dynamical Systems (DS) for motion representation with the capability to encode a family of trajectories rather than a fixed, time-dependent reference. A novel framework for task representation using DS is introduced, termed Locally Modulated Dynamical Systems (LMDS). LMDS differs from existing DS estimation algorithms in that it supports non-parametric and incremental learning while guaranteeing that the resulting DS is globally stable at an attractor point. To combine the advantages of DS motion

generation with impedance control, a novel controller for tasks described by first order DS is proposed. The controller is passive, and has the properties of an impedance controller with the added flexibility of a DS motion representation instead of a time-indexed trajectory.

Keywords: Varying Impedance Control, Manipulation, Dynamical Systems

RÉSUMÉ

LES humains font preuve d'une capacité impressionnante à manipuler des objets sans les endommager, en contrôlant la force et la position de leurs mains ou outils. Traditionnellement, la robotique a favorisé la commande en position par rapport à la commande en force pour produire des mouvements rapides, précis et reproductibles. Pour élargir l'applicabilité des robots manipulateurs en dehors des environnements strictement structurés des cellules de travail industrielles, la commande en position est insuffisante. Les tâches qui impliquent un contact physique avec des objets dont les positions ne sont pas connues avec parfaite certitude exigent un système qui contrôle la relation entre les déviations en position et les forces. Ce problème est formalisé dans le cadre de la commande d'impédance, qui met l'accent sur l'interaction entre le robot et l'environnement. En adaptant les paramètres de l'impédance, le comportement du robot peut être adapté aux besoins de la tâche. Cependant, il est souvent difficile de préciser comment l'impédance devrait varier pour mieux convenir à ces besoins. En outre, une variation d'impédance imprudente peut impliquer un comportement instable même en l'absence de forces d'interaction.

Dans la première partie de cette thèse, nous nous penchons sur la problématique de comment choisir un mode de control en impédance variable adéquat, et ce pour chaque tâche donnée. Pour ce faire, nous développons une interface homme-robot permettant à un superviseur humain d'enseigner des variations d'impédance en interagissant physiquement avec le robot pendant l'exécution de la tâche. Un set d'expérimentations robotique démontrent comment l'interface peut être utilisée pour améliorer la performance de plusieurs tâches de manipulation. Ensuite, nous nous penchons sur le problème de stabilité dans le contexte de la commande en impédance variable. Une discussion théorique sur ce sujet est présentée, ainsi que des conditions garantissant un comportement stable avec des profils variables de rigidité et d'amortissement.

Dans la deuxième partie de cette thèse, des scenarios de manipulation plus complexes sont explorés via la génération de trajectoires pendant l'exécution de la tâche. Cela se fait selon deux axes 1) apprendre à réagir aux forces d'interaction dans les tâches d'insertion qui sont essentielles pour des opérations d'assemblage et 2) représentation du mouvement par des systèmes dynamiques autonomes avec la capacité de décrire une famille de trajectoires en lieu d'une

seule trajectoire fixe. Un nouveau cadre pour la représentation des tâches par des systèmes dynamiques est introduit, qui permet de modifier un système dynamique existant par l'introduction de modulations locales dans l'espace d'état. Cette méthode porte l'acronyme LMDS (en anglais Locally Modulated Dynamical Systems). LMDS se distingue des autres systèmes d'apprentissage de systèmes dynamiques en ce que l'apprentissage non-paramétrique et incrémental est possible tout en garantissant que le système dynamique reste globalement stable à un point d'attraction. Finalement, pour combiner les avantages de la représentation de mouvement par des systèmes dynamiques et la commande en impédance, un nouveau régulateur pour des tâches décrites par des systèmes dynamiques est proposé. Ce régulateur est passif, et possède certaines propriétés avantageuses des régulateurs d'impédance avec la flexibilité supplémentaire d'une représentation de mouvement d'un système dynamique au lieu d'une trajectoire fixe et dépendante de temps.

Mots Clé: Commande en impédance variable, Manipulation, Systèmes Dynamiques

To Nora

ACKNOWLEDGMENTS

I would like to express my sincerest gratitude to Aude Billard. First of all, for giving me the opportunity to join the lab and pursue my PhD studies under her supervision. During these years, I have experienced an excellent balance between autonomy and guidance — I was encouraged to pursue my own ideas but always had feedback and support when I needed it. Thanks for your patience when I have been obstinate, all discussions, encouragement and support. Thanks also for many hours spent reading my draft reports, papers and thesis and for always giving invaluable feedback.

I would like to thank the members of my committee, Jonas Buchli, Etienne Burdet and Stefano Stramigioli, for very insightful, to-the-point and constructive feedback on my thesis draft. And for granting me the PhD title :) Thanks also to Jamie Paik who served as president of my committee.

When I first came to the lab it was to do a short summer internship. It was supervised by Mohammad Khansari. I enjoyed working with him so much that I extended the internship to be my masters project. Which I enjoyed so much that I ended up as a PhD student in the lab! Hence, Mohammad is very much responsible for this thesis work even having been started and for this I am extremely thankful. I have also been fortunate to continue to collaborate with Mohammad during my PhD. Mohammad has become a close friend, and I would also like to thank him for all the non-work related fun we shared during these years!

I would like to thank Sahar El Khoury for being a great friend and (almost) always distributing positive energy around you during these years. Thanks to Nicolas Sommer, for all the fun travels we have shared, and for organizing memorable evenings on the Lemman shore with Alsatian wine and Cuban cigars! Thanks also to Merve and Soha for lots great shared moments in these past years. Long live the Portugal gang! Thanks to Ajay Tanwani, for enjoyable office sharing and many fun moments.

Thanks to my office buddy Guillaume de Chambrier for policing my C++, scientific discussions and for all interesting, uninteresting, short, long, meaningful and meaningless discussions and bets we have had during this time. I would like to thank Basilio Noris, for always being helpful with coding issues and for being such an energetic and fun coworker and friend. Thanks to Eric

Sausser, whose support on everything from control to coding was invaluable in my first year in the lab. Thanks to Ashwini Shukla, the boss of debugging, for always giving a helping hand when needed and for all the fun during these years. Thanks to Sylvain Calinon for good advice and crazy times in conferences :) I would like to thank the old LASA generation who welcomed me in my first year, Brenna, Dan, Florent, Elena, Jean-Baptiste, Martin, Seungsu. The mid-generation for being great coworkers during this time, Miao, Lucia, Silvia, Luka, Hang, Ravin, Guillaume Pihen, Suphi, Joel. Finally thanks to the newest generation Sina, Nadia, Mahdi, Iaason, Laura, Felix, Joao, for being such a fun and energetic group!

I would like to thank my parents Ulla and Torbjörn for their support and encouragement. They have contributed to this thesis very concretely not only by conceiving its author, but also importantly by helping our little family when time has been short during these years. Thanks also to all family and Swedish friends who have injected energy and lots of fun by visiting us frequently, Joel, Amanda, Kristoffer, Jonatan, Ingrid, Maria, Robin, Carolin, Aron, Albin, Karl-Anders, Patrice, Karin, Kent, Anders, Madeleine, Jonathan (second one), Amanda (second one), Jonatan (third one!), Anna, Joel (second one), Martin, Jessica, Johan, Henrik. I would also like to thank our friends in Lausanne for memorable times skiing, wine tasting, traveling, hiking and partying, Jeff, Ksenya, Matt, Dom, Henrike, Kevin, Alina, Clara, Sebastian, Maryna, Mariia. Thanks to Geneva friends Sarah and Ruskin for many great evenings, brunches and lunches.

Finally the best and most important part. I am eternally grateful to Elin. First, for being my wife. Second, for always supporting me no matter what, always knowing how to cheer me up, and just understanding me very well. Third, for giving me the second love of my life, our daughter Nora. Thanks to Nora, for always being able to put a smile on my face.

TABLE OF CONTENTS

1	Introduction	3
1.1	Motivation	3
1.2	Approach	4
1.2.1	Varying Impedance Control	4
1.2.2	Motion Modeling with Dynamical Systems	6
1.3	Main Contributions and Thesis Outline	7
1.4	Publications	7
2	Background	11
2.1	Related work	11
2.1.1	Force and Impedance Control: Overview and historical perspective	11
2.1.2	Variable Impedance Control	13
2.1.3	Learning from Demonstration: Overview and historical perspective	15
2.1.4	Learning and Adaptation of Impedance Parameters	17
2.1.5	Control Strategies for Peg-in-hole insertion tasks	21
2.1.6	Dynamical Systems: Overview and historical perspective	23
2.1.7	Incremental learning in Dynamical Systems	25
2.1.8	Control of Tasks described by Dynamical Systems	27
2.2	Technical preliminaries	29
2.2.1	Dynamical Systems	29
2.2.2	Stability Analysis of Dynamical Systems	32
2.2.3	Rigid Body Dynamics	34
2.2.4	Impedance Control	35
2.2.5	Nonlinear Regression	39
3	Interaction Interface for Compliant Skill Transfer	45
3.1	Introduction	45
3.2	Online Learning of Varying Stiffness	46
3.2.1	Stiffness Adjustment Based on Interactions	46
3.2.2	Experiments	49
3.2.3	Discussion	55
3.3	Learning Compliant Manipulation through Kinesthetic and Tactile HRI	57
3.3.1	Filtering the interaction signal	57
3.3.2	Stiffness Increase Based on Grasp Pressure	59
3.3.3	Task Learning	61
3.3.4	User Study	71
3.3.5	Results	73
3.3.6	Discussion	74
3.4	Conclusion	75

4	Stability Issues in Varying Impedance Control	79
4.1	Introduction	79
4.2	Problem Statement	81
4.3	Ensuring Stability	82
4.3.1	Stability Conditions on Stiffness and Damping Profiles . .	82
4.3.2	Validating impedance profiles	83
4.4	Evaluation	86
4.4.1	Simulations	86
4.4.2	Validation of real impedance profiles	87
4.5	Discussion and conclusion	90
5	Haptic Reference Adaptation for Insertion Tasks	93
5.1	Introduction	93
5.2	Problem Statement	93
5.3	Learning Haptic Reference Adaptation from Human Demonstra- tions	95
5.4	Trajectory Generation and Robot Control	96
5.5	Experimental Setup	97
5.5.1	Preparing for Insertion	97
5.5.2	Data Collection and Model Learning	97
5.5.3	Autonomous Task Execution	100
5.6	Results	100
5.6.1	Comparison of Adaptation Schemes	102
5.6.2	Generalization	104
5.7	Discussion and conclusion	105
6	Incremental Motion Learning with Locally Modulated Dy- namical Systems	109
6.1	Introduction	109
6.2	Locally Modulated Dynamical Systems	111
6.2.1	Formulation and properties	111
6.2.2	Illustrative Examples	114
6.2.3	Modulation by Rotation and Norm-scaling	115
6.3	Learning Locally Modulated Dynamical Systems	116
6.3.1	Training Data	116
6.3.2	Gaussian Process Modulated Dynamical Systems	117
6.3.3	Enforcing Local Modulation	117
6.3.4	Trajectory-based Sparsity Criteria	118
6.4	Evaluation	119
6.4.1	Handwriting Motions	121
6.4.2	Non-convergence in reshaped systems	122
6.4.3	Polishing Task Using Planar Periodic Motion	125
6.4.4	Cartesian Trajectory Modeling for Stacking Plates	126
6.5	Discussion and conclusion	129
7	Passive Interaction Control with Dynamical Systems	133
7.1	Introduction	133
7.2	Problem Statement	134
7.3	Proposed Approach	135
7.3.1	Selective energy dissipation with task varying damping . .	135
7.3.2	Tracking in conservative DS	136
7.3.3	Extension to non-conservative DS	136
7.3.4	Impedance Adjustment	138
7.3.5	Choosing \bar{s}	140

7.4	Simulations	141
7.4.1	Planar free motion	141
7.4.2	Behavior in unexpected contact	143
7.5	Robot Experiment	146
7.6	Discussion and conclusion	150
8	Conclusions	155
8.1	Contributions	155
8.2	Limitations and future work	156
	Appendices	161
	Appendix A Appendices for Chapter 3	163
A.1	Gaussian Mixture Regression with Parameterized Generalization	163
A.2	System Usability Scale	165
A.3	NASA Task Load Index	165
	Appendix B Proof of Theorem 4.1	167
	Appendix C Appendices for Chapter 7	169
C.1	Smooth coupling functions	169
C.2	Proof of Proposition 7.1	171
C.3	Proof of Theorem 7.1	171
	References	173
	Curriculum Vitae	188

INTRODUCTION

1.1 Motivation

Decades of research and development in robotic manipulation has led to an incredible impact scientifically, socially and economically. Large manipulators move with sub millimeter accuracy at daunting speeds and have since long begun to relieve the human workforce from repetitive, non-ergonomic and dangerous tasks. The cost and the capabilities of robotic manipulators have so far limited their use to large-scale industrial production facilities, e.g. the auto manufacturing industry. Most such applications are characterized by a precise and deterministic task description and accurate position control which aims at rejecting any external forces applied on the robot.

In stark contrast, humans are capable of manipulation in uncertain environments, and exhibit physical compliance when subjected to perturbing forces. This compliance is realized by elasticity in the muscles that drive our limbs. We can vary the compliance by co-contraction of opposing muscle pairs. In this way, the compliance can be varied between different tasks and also during tasks. Human studies show that learning impedance variations are an important part of mastering a manipulation task (Burdet et al., 2001; Selen et al., 2009).

While traditional robot manipulators are inherently stiff due to their drive-trains, there is currently a strong trend toward torque-controlled, light-weight robots. These robots have been developed for quite some time and have now reached the level of maturity required for real applications. The availability of this class of robots opens a wealth of new possible applications, both in the industrial sector but perhaps even more so in the service robotics sector which involves health care and assistance applications. However, without suitable control these arms are no more useful than their caged predecessors. Indeed, for a robot to perform useful work, the controller and the task model is at least as important as the hardware.

This thesis is concerned with task modeling and control for compliant manipulation tasks. The contributions are anchored in the larger Learning from Demonstration (LfD) paradigm, which formalizes the acquisition of skills based on demonstrations of the task rather than explicit programming. There has been active research in LfD for decades (Billard et al., 2008; Argall et al., 2009),

mostly addressing how tasks are learned in terms of kinematic constraints. In this thesis, we explore how in addition to demonstration kinematic information for a task, the expert also teaches the robot how to vary its compliance. In this part of the thesis, the impedance control framework (Hogan, 1985b) is used. While this allows a certain amount of adaptability and robustness through the specification of task-based impedance variations, the task execution is sensitive to *temporal* perturbations that result in the robot and task model being out of sync. This issue can be addressed by using feedback of the robot state in the task progression. The task description is thus turned into a Dynamical System (DS), and a host of potential issues as well as possibilities arise. DS have been used for modeling human motions for decades (Bullock and Grossberg, 1988; Schöner and Kelso, 1988). In robotics, DS are becoming increasingly popular to model tasks, thanks to their capability of compactly encoding rich interactive behavior (Beer, 1995; Schaal et al., 2003). In the second part of this thesis a novel formulation for DS, particularly well-suited for incremental learning, is introduced. Finally, a controller that combines the advantages of the impedance control framework and task specification with DS is devised.

1.2 Approach

The contributions of this thesis can be categorized broadly in task-based varying impedance control and dynamical systems. These areas are introduced with detail below.

1.2.1 VARYING IMPEDANCE CONTROL

Try extending your arm and hitting it with your other arm. You will notice that your extended arm 'gives in' as it is impacted. This phenomenon is referred to as physical compliance. Compliance has an important functional role in our daily lives. For example, it helps to protect us from unexpected impacts, which are smoothly absorbed rather than rejected with a high intensity impact. But perhaps more interestingly, compliance is a crucial aspect of many manipulation tasks. The reader is here invited to pause for another personal experiment. Try inserting a key into a door — it should be easy as it is a task that you have likely been performing regularly for years. Now try the same task while making your arm as stiff as possible. You will notice that it is considerably more difficult. This is an example where the chosen compliance has an impact on the task performance. Numerous studies show that learning how to vary the compliance constitutes an important part of learning many manipulation tasks (Burdet et al., 2001; Franklin et al., 2008).

Robots can be controlled to be compliant by using an *impedance* controller. In impedance control, the goal of the controller is not to minimize a feedback error as is the case e.g. in pure position or force control. Instead, the goal is to maintain a certain dynamic relationship, termed mechanical impedance or

simply impedance, between external forces and resulting velocity of the manipulator (Hogan, 1985b). In general terms, a high impedance means that the robot tries to insist on its position and will tend to reject external forces. A low impedance means a compliant behavior and the robot will tend to give in to external perturbations. Developments of hardware platforms on which impedance control can be easily implemented has lead to a renewed interest in the topic in recent years, with the hope that compliance can be as useful for robots as it is for humans.

Most applications if impedance control model the robot as a mass-spring-damper system attached to a moving reference point, called the virtual trajectory¹. The virtual trajectory is the path that the robot follows if it is left unperturbed. When robot encounters external forces or obstacles blocking its path, it will depart from the virtual trajectory and will be pulled back toward depending on the value of the *inertia*, *damping* and *stiffness* parameters which define the impedance of the robot. On an impedance controller implemented in software, it is straightforward to vary the impedance over time. Even some inherently compliant actuation system allow variation of the impedance (Albu-Schaffer et al., 2008). The question then rises how to exploit this capability, i.e. how to choose the impedance profile for a task. Humans are good at figuring out how to do tasks but rather bad at describing them in a way that is understandable to a robot. This fact is the motivation of the LfD paradigm, which aims to endow robots with the capability to learn tasks merely by observing demonstrations of it (Schaal et al., 2003)(Billard et al., 2008). In this thesis, LfD is extended to include compliant manipulation tasks. This is achieved by the introduction of a haptic teaching interface that allows the human to teach the robot stiffness variations by physically interacting with it during task execution. It is shown that this interface allows to teach robots stiffness variations that increase performance in several tasks. Furthermore, a user study confirms that lay users are capable of determining correct impedance and that the interface intuitively allows them to transfer this insight to the robot.

This thesis and numerous other works have shown the benefit of varying impedance according to the need of the task and devise various ways to define this varying impedance. However, it is possible to create impedance profiles that yield an *unstable* behavior of the robot. While this is rare in practice, it is important to better understand to what extent the impedance can be varied without risk of instability. This thesis addresses this problem by deriving bounds for variation of stiffness and damping. It is shown that if the stiffness and damping are varied within those bounds, the system can not go unstable in the contact free case nor in interaction with any passive environments.

As mentioned previously, the behavior of a robot under impedance control is determined by the impedance parameters and the virtual trajectory. In the lit-

¹Impedance control is not limited to this particular second order model, but this particular choice makes it easy to implement on robots and is for many applications sufficiently flexible.

erature, the virtual trajectory is often assumed to be a predefined trajectory — often numerically represented using splines (Lin et al., 1982) — that is replayed each time the task is executed. Perhaps surprisingly, this approach in practice often works well even for tasks with perturbations, as long as the perturbations don’t significantly delay the progression of the task. However, the use of a predetermined movement trajectory is by definition very limiting. More intelligent behavior can be achieved by online generation of the virtual trajectory. This thesis explores two directions that adapts the virtual trajectory during interaction control. First, the challenging insertion type of tasks are considered. The problem of inserting a peg into a hole is tackled by generating the virtual trajectory online, as a function of the forces experienced by the robot when trying to insert the peg. Second, Dynamical Systems (DS), are explored for use in compliant manipulation tasks.

1.2.2 MOTION MODELING WITH DYNAMICAL SYSTEMS

DS for motion modeling are a longstanding research topic in robotics and human motor science (Bullock and Grossberg, 1988; Schöner and Kelso, 1988; Schoner, 1990; Beer, 1995). The use of DS for representing motion has the advantage that the trajectory for the robot unfolds during the task, and can be defined as a function of different variables, e.g. the robot position, measured forces and time. Of particular interest are autonomous DS. An autonomous DS takes as input a state variable, e.g. the robot position, and returns a rate of change of that variable (velocity for the example of position as input). It has been demonstrated that many DS representations lend themselves well to learning, in supervised settings (learning from demonstration) as well as reinforcement learning. Furthermore, in some cases qualitative properties such as stable limit cycles or stability can be guaranteed by the formulation itself (Schaal, 2003), or by constrained optimization (Khansari-Zadeh and Billard, 2011; Neumann et al., 2013). Hence, these properties which are often easy to determine if they are desired in a task, can be ensured regardless of the type of data that is then provided to respective learning algorithm, as long as the constraints are respected. However, stability constraints have the side-effect of restricting the range of motions that can be modeled. If complex trajectories are given as demonstrations, the DS is often not capable of reproducing such motions. In this thesis, a novel DS representation is introduced which unlike previous methods for learning autonomous dynamical systems does not use constraint based optimization for learning. Instead, the DS is represented on a form that makes it impossible by construction to represent unstable systems. This formulation allows incremental learning, i.e. it can gradually learn from continuously arriving data points.

To illustrate the importance of DS for manipulation in uncertain environments, consider an example where a robot is moving its arm along a prede-

terminated, time-dependent path using an impedance controller. If the robot is obstructed along the path, the virtual trajectory keeps advancing while the robot stays put. The impedance controller tries to pull the robot toward the virtual trajectory, creating an increasing contact force between the robot and the blocking obstacle. This force can become a hazard for the robot or blocking object, and can result in a violent release of energy if the obstacle is suddenly removed. If the virtual trajectory was generated by an autonomous DS, there would be no increase in contact force since the progression of the virtual trajectory would depend only on the actual position of the robot. The idea of combining these two paradigms hence seems promising. The last chapter of this thesis introduces a novel controller that combines the advantages of the DS as a motion representation and impedance control for managing interaction with the environment. The controller is *passive*, which means that in addition to being stable in free motion, it is also stable in interaction with any passive environment (van der Schaft, 2000; Ortega et al., 1998).

1.3 Main Contributions and Thesis Outline

A brief list of the main contributions in the order that they appear in the thesis is given here. In Chapter 3, we introduce a novel **teaching interface for varying impedance** allowing users to interactively update the impedance of the robot during task execution and with haptic feedback. With the possibility of teaching and using varying impedance in the task model, it is important to ensure that the impedance variations do not yield unstable behavior. This topic is treated in Chapter 4 which introduces **sufficient conditions for stability for varying stiffness and damping profiles**.

While Chapter 3 and 4 focus on impedance control with time-dependent virtual trajectories, the Chapter 5 introduces online trajectory modification based on sensed forces for insertion tasks. Chapter 5 uses existing methods and does not contain any algorithmic novelty, but rather innovates in the way that existing tools in compliant control and learning from demonstration can be used in combination for insertion tasks.

Chapter 6 introduces a novel DS formulation especially well suited for incremental learning, termed **Locally Modulated Dynamical Systems (LMDS)**. A learning algorithm based on Gaussian Processes is also presented. Chapter 7 is focused on devising a controller that can incorporate LMDS and similar DS motion representations into an interaction control framework. A novel controller named **Passive Interaction Control with Dynamical Systems (PICDS)** is introduced, formalized and evaluated. The final chapter provides a summary and recap of limitations and possible future directions of research.

1.4 Publications

Large portions of this thesis have been published in peer-reviewed conferences and journals. The contents of Chapter 3 has been published in (Kronander and Billard, 2012) and (Kronander and Billard, 2013). The work on stability conditions for varying impedance in Chapter 4 is at the time of this writing under review for publication in a robotics journal. The LMDS formulation of Chapter 6 is published in (Kronander et al., 2015). Chapter 5 has not been published, nor has Chapter 7. The latter is at the time of this writing in preparation for submission in a robotics journal.

The author has also authored/coauthored a number of publications which are not part of the thesis. In the authors master thesis, the application of DS models for hitting motions in mini-golf was explored (Kronander et al., 2011). Among our efforts at a closed loop DS formulation (the topic of Chapter 7) was a constraint based approach presented in (Khansari-Zadeh et al., 2014), which uses a constrained second order DS in an effort to unify impedance control and motion generation. Recently, an internship project supervised by the author was published in (Cha et al., 2015). In that paper, the DS approach of Chapter 6 of this thesis was used in a HRI study investigating interfaces for teaching motions to robots.

BACKGROUND

2.1 Related work

In this section, I review relevant literature organized along three main axes: Impedance control, LfD and DS. The first part of the thesis is concerned with developing methods for teaching robots varying impedance information. Hence, it is related to and builds upon works from both impedance control and LfD. A general overview of these fields are given in Sections 2.1.1 and 2.1.3. A focused review related to the challenges addressed in Chapters 3 and 4 are given in Sections 2.1.4 and 2.1.2. Discussion on related works for the peg-in-hole problem treated in Chapter 5 is given in Section 2.1.5. A general overview and historical perspective on DS is given in Section 2.1.6, and Sections 2.1.7 and 2.1.8 zoom in on the topics of Chapters 6 and 7.

2.1.1 FORCE AND IMPEDANCE CONTROL: OVERVIEW AND HISTORICAL PERSPECTIVE

Controlling robotic manipulators in physical contact with the world remains a difficult problem. Researchers have tackled this problem for decades, with a wide spectrum of control algorithms and even specialized hardware design as a result. The motion control problem — which deals with controlling a robot *without* contact — is today largely considered a solved problem where the research is focused on optimizing the well-established frameworks of inverse dynamics control, adaptive motion control and independent joint PID control (Slotine, 1987). In contrast, only in the last decade has the research started to converge to a particular class of control in the complementary field of control *with* physical contacts. This class is impedance control (Hogan, 1985b), which has emerged as suitable control architecture to improve performance of robotic manipulation in many situations where traditional control paradigms are prone to failure. Examples include tasks with uncertain pose estimates of manipulated objects, physical contact and tasks in which it is important to respond appropriately to unforeseen perturbations (Villani and De Schutter, 2008). Before going further, it is worth briefly reviewing the state of compliant control algorithms at the time of Hogans introduction of impedance control. Adaptation of robot trajectories based on force feedback had been investigated (Whitney, 1977). The conflict be-

tween simultaneous control of position and force inspired (Mason, 1981) which introduced the idea of separating the work-space in orthogonal position- and force-controlled sub-spaces. The hybrid control approach (Raibert and Craig, 1982) belongs to the same category of control. While these approaches could be successfully applied in many scenarios, they have issues in terms of performance mainly stemming from non-consideration of the manipulator dynamics (An and Hollerbach, 1989; Whitney, 1985). This is addressed in the operational space formalism (Khatib, 1987), but the geometric task specification still needs high accuracy and is tedious to establish. It has later been shown that the classical hybrid control is not well posed when considering motions including change of orientation (Brockett, 1993; Lipkin and Duffy, 1988). The parallel approach to position and force control (Chiaverini and Sciavicco, 1993) was introduced to overcome the shortcomings of the hybrid controllers, in this case by effectuating simultaneous feedback control of position and force on all axes but giving priority to force control by integral control action. Instead of trying to control forces explicitly, the stiffness control method (Salisbury, 1980) allowed the task designer to specify a desired trajectory and a corresponding desired stiffness. This technique did not require the orthogonal task spaces and can be implemented without a force sensor on torque-controlled mechanisms. Desired contact forces can be achieved by deliberately offsetting the reference trajectory inside the object to contact. In the quasi-static case, the relationship between external forces and the deflection of the end-effector is given by the stiffness value. Hogan generalized this idea and modeled the relationship between physical *effort*(force) and *flow*(velocity) as a second order linear dynamical system, parameterized by a virtual inertia, damping and stiffness (Hogan, 1985b). Fig. 2.1 gives a simple illustration of the differences between the hybrid, parallel and impedance control paradigms.

Rather than a particular control implementation, impedance control should be viewed as a concept that can be implemented in various ways. The implementation scheme suggested in (Hogan, 1985a) applies to torque-controlled manipulators and reassembles the work of (Salisbury, 1980). For manipulators with high gear ratios, e.g. most available industrial manipulators, it is not possible to implement impedance control in this manner. Instead, the causality of the impedance can be reversed by using force measurements to simulate a trajectory according to the desired impedance model. This trajectory is then fed to a high performance position controller. These two alternatives of implementation are usually referred to as impedance control and admittance control respectively. The impedance causality implementation has better robustness in rigid contact, but the admittance control scheme is more widely applicable and has better nominal performance in free motion (Ott et al., 2010). For the implementation with impedance causality, the ability to command the torques to the robot joints is necessary. In the 80's and 90's, platforms on which this was possible only existed as custom built prototypes in universities around the

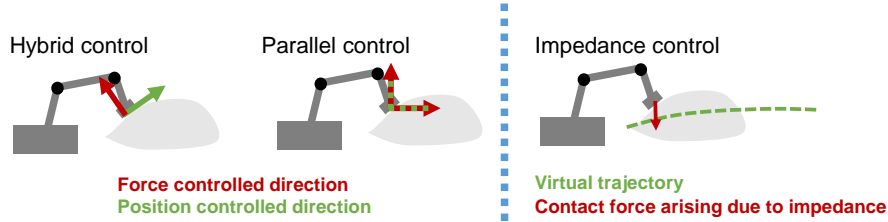


Figure 2.1: Conceptual illustration of the hybrid, parallel and impedance control approaches.

world. Since then, a number of robotic manipulators that allow accurate torque control have been introduced. A notable example is the work of Albu-Schaffer et al. which exploited the use of link side torque sensors to allow high performance torque control which could then be used to implement impedance control (Ott et al., 2004; Albu-Schaffer et al., 2007b,a). This research resulted in the development of the KUKA LWR family of robots which are now widespread in the academic community and also increasingly in industrial applications.

A challenging alternative to implementation of impedance control through control as described above, is to physically design the robot such that it obeys the dynamics of the target impedance model. Ideas of exploiting passive compliance in the kinematic structure have been present for a long time, including e.g. the remote center of compliance (Drake, 1977; Whitney and Nevins, 1979; Loncaric, 1987; Brockett, 1993). In recent years, compliant actuation has been an extremely active field of research, pioneered by (Pratt and Williamson, 1995) who introduced the series elastic actuator (SEA) which has seen numerous commercial applications. While the SEA has a fixed compliance, various designs that allow adjustment of the compliance have emerged in the last decade (Migliore et al., 2005; Tonietti et al., 2005; Bicchi et al., 2005; Ham et al., 2009). Compliant actuation offers many of the promised benefits of impedance control — shock resistance, stable contact with stiff environments and safe human-robot interaction to name a few — in a more robust and safe version than can be offered by the active impedance control algorithms reviewed previously (Albu-Schaffer et al., 2008). However, there are also a number of drawbacks, the most important being that the control problem becomes considerably more difficult (Palli et al., 2008) and the range of impedances that can be rendered are usually very limited. Furthermore, with few exceptions (Catalano et al., 2012; Laffranchi et al., 2011), soft actuator design has focused on implementing variable stiffness but fixed damping. In practice this generally results in significantly under- or over-damped systems with poor tracking performance as a consequence. For a recent review on research in variable impedance actuation, the reader is referred to (Vanderborght et al., 2013).

2.1.2 VARIABLE IMPEDANCE CONTROL

While impedance control with constant impedance parameters is already an adequate solution in some cases, varying the impedance parameters during the task gives more flexibility and can drastically impact performance in many tasks. The match-lighting task in Section 3.3.3 of this thesis is a good example of a task that could not successfully be executed without varying the impedance. In biological motor control, it is known that humans use varying impedance in a task-dependent manner (Burdet et al., 2001; Gomi and Osu, 1998). As reviewed separately in Section 2.1.4, significant research efforts have dealt with and are still dealing with the assignment of varying impedance to achieve various goals, e.g. better performance, higher safety or lower energy consumption. A topic that has been largely ignored by this community are the stability implications of letting the impedance parameters vary. Exceptions include works that propose impedance variation according to specific adaptation laws. A notable example is (Yang et al., 2011), in which a biologically inspired varying impedance controller was proposed and its stability rigorously proven. For the general case where the impedance can be considered time-varying, it is easy to show that the standard energy-based stability analysis (Takegaki and Arimoto, 1981) is no longer valid in the case of an arbitrarily varying stiffness matrix (this is done in Section 4.2 of this thesis).

Variable impedance control can be seen as a particular case of gain scheduling, which is a common technique in control for applying linear control design techniques to nonlinear systems¹. In gain scheduling, it is generally not possible to theoretically guarantee stability. Instead, the approach usually involves practical rules-of-thumb and extensive numerical simulations for assessment of stability (Rugh and Shamma, 2000). Typically, stability can be ensured only locally and under rather vague conditions such as “slowly varying” gains (Shamma and Athans, 1992). The success that variable impedance control has already had in numerous applications is an indication that reasonable impedance variations are sufficiently small/slow for a wide range of tasks. Nonetheless, these notions are of limited use when designing varying impedance profiles since there is no way of knowing how slow or how small the variations need to be.

Time-domain passivity is based on computing energy quantities online and injecting damping as necessary to ensure stability. The concept was used in (Hannaford and Ryu, 2002) for haptic interfaces, in (Ryu et al., 2004a) for tele-manipulation and generalized to more control scenarios in (Ryu et al., 2004b). This approach is based on a passivity observer providing information to a passivity controller responsible for injecting damping whenever the passivity observer reports positive energy generation. The passivity layer (PL) approach introduced in (Franken et al., 2011) also uses time-domain passivity for dealing with interactivity and delay in bilateral tele-manipulation. It is based on the pos-

¹Essentially, the approach of gain scheduling consists in 1) linearizing the nonlinear system around a set of operating points, 2) Designing feedback controllers for each of the linearizations using linear control tools and 3) Controlling the nonlinear system by interpolating between the feedback gains found for the linearizations.

sibility to compute exactly the energy exchange between an environment and a robot, which as was shown in (Stramigioli et al., 2005) is possible even for sampled system provided the robot has impedance causality (velocity in, force out).

An approach similar to the time-domain passivity ideas discussed above was introduced specifically for the case of varying stiffness matrix in impedance control in (Ferraguti et al., 2013). Their controller also monitors online the energy of the system but stabilization is based on falling back to a constant stiffness component rather than introducing additional damping. As is shown in Chapter 4 of this thesis, the approach of (Ferraguti et al., 2013) yields discontinuities in the stiffness profile which is usually not desirable for control of real robots.

While the methods discussed above could theoretically stabilize a system under impedance control with an arbitrary varying stiffness profile, the way that it achieves this is by either injecting damping (Ryu et al., 2004b) or by modifying the stiffness profile (Ferraguti et al., 2013) depending on the state trajectory observed *during task execution*. It is hence impossible to determine the impedance profile that the robot will use before task execution. Consequently, an engineer carefully having crafted a task-specific impedance profile could end up watching in frustration as the robot exhibits a totally different behavior than intended as it performs the task.

Chapter 4 of this thesis derives sufficient conditions for stability for varying stiffness and damping profiles. In contrast to the approaches reviewed above, these conditions are *state-independent* and can hence be evaluated before task execution. A comparison to the approach of (Ferraguti et al., 2013) is also provided.

2.1.3 LEARNING FROM DEMONSTRATION: OVERVIEW AND HISTORICAL PERSPECTIVE

The idea of endowing robots with the capability to learn from task demonstrations has existed for decades. Early developments in the 80's focused on developing methods to automatically synthesize robot programs from demonstrations (Lozano-Perez, 1983; Levas and Selfridge, 1984; Dufay and Latombe, 1984; Segre and DeJong, 1985). At this stage, the field was focused on reducing a task into a symbolic graph-like structure that could be traversed using simple if-then rules (Billard et al., 2008; Schaal, 1999).

While the symbolic approach was successful in many tasks and is still an ongoing research topic, the those early works synthesized programs with limited flexibility. Enter the trajectory-based approach to LfD, which by applying powerful tools from statistical learning theory and artificial neural networks allows to capture tasks in a more detailed way. Many works have focused on encoding trajectories as time-dependent functions using e.g. splines (Ude, 1993; Delson

and West, 1996; Ude et al., 2004) or nonlinear regression techniques (Calinon et al., 2007; Shon et al., 2005). One of the central concepts in trajectory-based learning from demonstrations is consolidation of multiple demonstrations that may be corrupted by a large amount of noise and that can generally not be expected to be synchronized. The latter problem is often handled using Dynamic Time Warping. As an alternative, Hidden Markov Models (HMM) were adopted for dealing with multiple demonstrations without the need of a preparatory synchronization step (Tso and Liu, 1996; Lee and Xu, 1996). Task representation using Dynamical Systems (DS) was identified early as a powerful means of dealing with the same difficulties, and at the same time providing a number of advantages at task execution time. The DS approaches are reviewed separately in Section 2.1.6.

While the majority of works in Learning from Demonstration have focused on learning kinematic information only, forces to apply to the environment can be equally important. Asada and Izumi were very early in addressing compliant manipulation tasks for LfD in (Asada and Izumi, 1989). At the time, the control algorithm that was best understood and most widespread for contact tasks was the hybrid force/position controller (refer to Section 2.1.1). The authors use position and force data to automatically program the position and force set points and in addition piece-wise static selection matrices required in the hybrid control framework. The approach naturally inherits all the problems of the explicitly programmed hybrid approach and in addition needs to deal with task segmentation and hard control switching. More recent works force-based LfD belong to the trajectory-based family of approaches include (Rozo et al., 2011; Kormushev et al., 2011; Koropouli et al., 2012). Simultaneous control of force and position is physically impossible (Stramigioli, 2001), and therefore these approaches all have to make compromises at execution time. An alternative is to instead focus the learning on the impedance of the robot, which is reviewed separately in Section 2.1.4.

Teaching interfaces in LfD have ranged from tediously moving robot via set points, vision (Kuniyoshi et al., 1994; Kang and Ikeuchi, 1995; Tung and Kak, 1995) to haptic devices (Kormushev et al., 2011) and kinesthetic teaching (Calinon et al., 2006), see Fig. 2.2. The latter is possible on backdrivable robots and is today probably the most popular interface for providing demonstrations. It can be used to capture kinematic trajectories and if force-sensors are mounted between the point of contact with the teacher and the tool, contact forces can also be captured in this manner. The situation is different when we wish to transfer impedance information from the teacher. Unlike position or force, impedance is not something directly measurable that can be easily captured by demonstration. A number of solutions have been proposed to resolve this issue, and these are reviewed separately in Section 2.1.4.



Figure 2.2: An example of kinesthetic teaching. The robot is gravity compensated, allowing the teacher to effortlessly guide it through the motion.

2.1.4 LEARNING AND ADAPTATION OF IMPEDANCE PARAMETERS

LEARNING FROM DEMONSTRATION

Until a few years ago, LfD was with few exceptions (Asada and Izumi, 1989; Asada, 1993; Sikka and McCarragher, 1997) focused on the representation and learning of kinematic tasks. Hence, task parameters such as desired motion trajectories were learned, and for executing the task autonomously after learning, the robot typically relied on low level position controller. For tasks involving free motion and simpler tasks involving contact, this is a proven approach which has been employed in industrial robotics for decades. However, for tasks requiring specific forces to be exerted on the environment, or tasks involving contact as important elements, such an approach is prone to failure. Impedance control (see Section 2.1.1) can handle such situations, but require additional task-related information: the specification of the impedance. This is even more important if the impedance should be allowed to vary, which is generally required to achieve best task performance. This fact is the motivation for a number of recent works in LfD and for Chapter 3 in this thesis.

In (Sikka and McCarragher, 1997), it was proposed to use human demonstrations to derive data for the stiffness control framework (Salisbury, 1980). The authors realized that trajectory information is insufficient to capture the human manipulation skills, and propose a set of constraints aimed at facilitating the identification of desired trajectory and stiffness parameters from recordings of positing and force of human manipulation tasks. However even with these constraints the desired trajectory and stiffness are not uniquely identifiable and strong heuristics are used in the identification procedure.

More recently, Calinon et al. proposed to derive stiffness variations for a compliant controller from the kinematic demonstrations of the position trajectories in (Calinon et al., 2010). A probabilistic model is fit to the demonstrated trajectories, and the stiffness profile is shaped so that the robot adopts a high stiffness in directions of low variance. In (Kormushev et al., 2011), the same

approach for setting the stiffness as in (Calinon et al., 2010) was used. Furthermore, the robot learned a model of the contact forces that should arise in correct manipulation. First, the positional profile of the task was taught using kinesthetic teaching. When the positional profile for the task was learned, the teacher demonstrated the interaction forces using a haptic device while the robot was executing the learned motion. A model describing the desired contact forces was then built, and used by the robot to determine the desired force during task execution. The robot was then controlled by adding the desired contact force to the output of the compliant position controller. The approach for deriving the impedance profiles in (Calinon et al., 2010) and (Kormushev et al., 2011) is based on the assumption that the user conveys impedance information through variability of the demonstrated motion. The compelling aspect of the approach is that impedance information is derived from kinematic data, which is necessary anyway, and the user is hence not required to demonstrate also the impedance information. The main disadvantage is that the assumption that impedance is directly related to kinematic variability in the demonstrations may not always be true, and the user may end up with a robot using impedance variations that he/she did not wish to convey.

In (Medina et al., 2013), a risk-sensitive Linear Quadratic Regulator (LQR) control strategy is proposed for position and force trajectories encoded with covariance information. Demonstrations are taken into account in computing the optimal control, by employing the inverse covariance of the demonstrated trajectory data as metrics when computing the cost. The activity of a force feedback controller is considered as process noise to a trajectory tracking controller, which results in stiffness reduction during significant force correction. Additionally, the force control is facilitated by an online adaptation of the reference trajectory depending on the force feedback error. A similar technique has been used for haptic assistance in human robot collaboration scenarios in (Medina et al., 2015). The derivation of a varying stiffness and damping matrices from kinematic data has also been formalized in the context of minimal intervention control via finite horizon LQR in (Calinon et al., 2014). Much like these approaches (Lee et al., 2015) uses demonstration data for deriving a time varying stiffness and damping matrices. In that case, each dof is considered separately (corresponding to a stiffness matrix that is constrained to be diagonal), and priors are assigned to the stiffness and damping matrices which are estimated in Bayesian framework. A similar technique, without priors on the stiffness matrix and instead a post-estimation projection onto the admissible space of symmetric positive definite matrices was proposed in (Rozo et al., 2013).

In Chapter 3, we take a different approach and develop a user interface that allows the teacher to communicate the desired stiffness variations explicitly. The system is based on interpreting physical perturbations by the teacher on the robot as an indication to locally reduce the stiffness. Our approach maps variability in the interaction to reduced stiffness in our teaching interface and

is in that aspect similar to (Calinon et al., 2010) and (Kormushev et al., 2011). The stiffness profile is however learned independently and does not depend on the positional demonstrations. An important difference is that in our approach, the user gets haptic feedback during teaching, since the robot adapts its stiffness online. Not having this feedback would be comparable to providing kinematic demonstrations without visual feedback. The approach of demonstrating stiffness in this way has later been applied in the context of object level impedance control for in-hand manipulation in (Li et al., 2014).

REINFORCEMENT LEARNING AND OPTIMAL CONTROL

Learning variable impedance control policies has been formulated as an optimal control problem (Mitrovic et al., 2011),(Braun et al., 2012). These works specify the task constraints as a cost function and optimize the control actions subject to the dynamics of the robot. This has the advantage that the impedance profile is tailored to each robotic platform. Solutions based on optimal control theory for varying the stiffness to maximize link velocity of VSA systems have been reported by (Garabini et al., 2011) and (Haddadin et al., 2011). The optimal control approach is especially well-suited for highly dynamic tasks that need to exploit the passive dynamics of compliantly actuated systems (Braun et al., 2013).

Closely related to optimal control is reinforcement learning (RL), which was first used for learning variable impedance policies in (Buchli et al., 2010) and (Buchli et al., 2011). The approach is based on formulating the varying stiffness as a differential equation that becomes an additional state in the Dynamic Motor Primitive (DMP) framework (refer to Sections 2.1.6 and 2.1.7). Model-free reinforcement learning (Theodorou et al., 2010) is subsequently used to learn all the parameters of the extended DMP, hence simultaneously obtaining the reference trajectory and the stiffness profile. Interestingly, the approach has been able to qualitatively reproduce patterns of impedance learning observed in humans (Stulp et al., 2012).

Kormushev et al. used a control policy similar to DMP but where the non-linear forcing terms are replaced by a set of spring damper systems centered on points along the path of the task (Kormushev et al., 2011). It differs from (Buchli et al., 2011) in that the acceleration output from the DMP is directly used as a control command via inverse dynamics, without a separate PD controller. An EM-based reinforcement learning algorithm initialized by human demonstrations is presented, which was demonstrated in a pancake flipping task refined through trial-and-error. Improvement of the EM-based reinforcement learning algorithms allowing multi-optima search were presented in (Calinon et al., 2013).

All works in reinforcement learning and optimal control rely on a well designed cost function that is able to capture quantitatively the performance of a trial of the task. With well designed cost functions, the works cited in this section have had formidable success at learning very challenging tasks. The spec-

ification of a cost function for general tasks is however a daunting task, even for experts (Schaal et al., 2003). Important research is being done to facilitate the procedure by estimating a cost function from demonstrations through inverse reinforcement learning (Abbeel and Ng, 2004). In particular, this approach has been used in the context of transferring impedance modulation strategies between heterogeneous systems in (Howard et al., 2013). So far, these algorithms rely on predetermined feature selection, and are hence limited in their applicability for lay users. RL and LfD hence continues to coexist as complementary approaches to task acquisition, with an increasing amount of cross-breed as exemplified by (Calinon et al., 2013).

BIOLOGICALLY INSPIRED APPROACHES

Humans and other animals are great examples of control systems that leverage strongly on compliance. A large body of research — to vast for complete coverage here — seeks to understand how humans vary impedance. Mechanically, the stiffness of a human arm is determined by the level of co-contraction of opposing muscle pairs (Won and Hogan, 1995; Rosenbaum, 2009). An apparatus for measuring human limb stiffness was presented in (Gomi and Kawato, 1996) and used for examining hypothesis regarding human arm movement control. Similar setups have since then been used in a number of experiments, including (Gomi and Osu, 1998) which discusses task-dependent stiffness in humans, and (Burdet et al., 2001) which reports that humans modulate stiffness to deal unstable task dynamics. Similar evidence for task-based adaptation of impedance in humans have been reported in a number of works (Franklin et al., 2008; Takahashi et al., 2001; Wang et al., 2001). A controller that adapts position trajectory, feed-forward force and impedance according the principles derived from those human studies was presented in (Yang et al., 2011). The adaptive behavior of this controller is to use feed-forward force to compensate for predictable perturbations, whereas unpredictable perturbations are tackled by learning to selectively increase stiffness in directions where stochastic perturbations are applied. An extension to this controller including reference adaptation for allowing force control and haptic exploration was presented in (Ganesh et al., 2012).

The works cited above aims at understanding how humans modulate impedance and in some cases uses results from these studies to develop robot controllers. In contrast, our work does not aim at modeling the impedance variations that the human would use if he did the task with his own arm, but rather uses the human as a teacher who is assumed capable of teaching an impedance profile that is suitable *for the robot and the task at hand*. Among the reasons for impedance modulation in humans are readiness to reject perturbations (Burdet et al., 2001), dealing with sensorimotor noise (Selen et al., 2009) and gradual stiffness decrease to reduce energy expenditure (due to muscle fatigue). In place of replicating the process that humans use to modulate impedance, we aim at modeling solely the aspect that is related to the task, which may be different

for the robot and the human due to different structure and mass properties of the arm etc.

Among the tools in studying human impedance modulation is surface EMG (Osu and Gomi, 1999). This has been exploited to implement “tele-impedance” in (Ajoudani et al., 2012). On the master side, the hand position of the operator is tracked with a visual marker system, and an online estimation of the stiffness using surface EMG is performed simultaneously. The hand position along with the estimated stiffness is then send to the slave side, where the robot under impedance control updates its reference position as well as its stiffness to that values send from the master. With the user only receiving visual feedback from the slave side, the question rises whether the user uses the same stiffness she would if she was doing the task directly, or if she rather uses the system as an interface to select a stiffness that seems appropriate for the robot. While the authors do not answer that question in (Ajoudani et al., 2012), they do report several example tasks where tele-operation using their system outperformed tele-operation with constant low or constant high stiffness on the slave side. Depending on the role the user takes with the system proposed in (Ajoudani et al., 2012) this may be the work which is most closely related to Chapter 3 of this thesis as it uses an interface to set the stiffness explicitly.

2.1.5 CONTROL STRATEGIES FOR PEG-IN-HOLE INSERTION TASKS

Insertion tasks are a major difficulty for automatizing manufacturing processes. The bench-mark problem in this category, peg-in-hole, has been studied for decades and is representative of the challenges that arise in uncertain assembly operations. The difficulty of the peg-in-hole task and its importance for assembly operations in the manufacturing industry are well documented in the literature (Hannaford et al., 1991). If the pose of the parts can be tracked with negligible inaccuracy, the problem can be solved by generating trajectories based on the geometry of the parts (Lozano-Perez et al., 1984). Here, we review research related to Chapter 5, which considers the more difficult case were the uncertainty of the pose estimate or accuracy of control is significantly larger than the clearance. Insertion operations were also the motivation behind the development of the Remote Center of Compliance RCC device (Drake, 1977). For the RCC to work well, the center of compliance should be placed near the tip of the peg (Whitney, 1982), which may require a different device for different objects. RCC that allow to control the location of the center of compliance have been proposed Cutkosky and Wright (1986). Here, we instead solve the insertion problem not with an auxiliary device but rather via actively actuating the robot to complete the insertion.

In (Asada, 1990, 1993), it was proposed to use multilayer neural nets mapping sensed end-effector force to desired velocity as a representation of non-linear

compliance. The neural nets were trained by minimizing the Mean Square Error (MSE) of a set of training data consisting of pairs of force and corrective velocity. The availability of a sufficiently large set of noise-free training data was assumed.

Reinforcement learning has been employed in (Burdet and Nuttin, 1999; Gullapalli et al., 1994) for gradually acquiring a map from sensed force to corrective velocity for peg-in-hole insertion. In reinforcement learning, the robot is evaluating its own attempts at the task using a task-specific cost function. A general problem with applying reinforcement learning to real robots is that a large amount of trials is usually needed for learning the task. Both (Burdet and Nuttin, 1999) and (Gullapalli et al., 1994) reported hundreds of trials before the task had been successfully learned. For interaction tasks, this is especially problematic since hazardous contact forces can arise during the trials (Burdet and Nuttin, 1999).

Our solution for learning how to adapt the reference trajectory based on the sensed wrench is similar to the works discussed above in how we model the reactivity: sensed wrench in, velocity out. However, in contrast to (Asada, 1993) we use a probabilistic model of the demonstrated data and propose a viable means for collecting such data (which was absent from (Asada, 1993)). We differ from (Burdet and Nuttin, 1999) and (Gullapalli et al., 1994) in that we use LfD rather than reinforcement learning, which are to be considered complementary approaches that are suitable in different situations (see also discussion in Section 2.1.4).

An easily implementable approach to peg-in-hole is to use slight random movements or oscillations of the reference trajectory (Badano et al., 1991). With this approach, the idea is that the misalignment will eventually be compensated by pseudo-random movements applied to the end-effector. To limit the contact forces that will obviously arise when the reference is adapted in the wrong direction, it is necessary to combine this approach with a lower level compliance. Our approach in Chapter 5 should be seen as an intelligent variant of such a scheme: randomization is used but it depends on the sensed wrench.

Passive compliance has been exploited for peg-in-hole in (Balletti et al., 2012). A hand-crafted search strategy was used, completing the insertion via aggressive thrusts. This work investigates how to compliantly actuated systems can be exploited for such aggressive search. However, the approach results in comparatively high contact forces and the insertion strategy is fully hand-coded. A related work is (Park et al., 2013), which also presented a hand-coded search strategy for peg-in-hole insertion. Predefined oscillatory motions were used which combined with lower level compliance eventually performed the insertion successfully in all reported experiments. No report on the contact force experienced during is given, and the program is hand-coded for a specific peg-in-hole insertion. In that aspect it is more similar to industrial practice where there are numerous examples of assembly applications in the same range of tolerance

or smaller, with close perfect performance in terms of success rate and with significantly lower completion times. These applications are always carefully hand-crafted for a specific application, often with task-specific tools and always with a very low amount of uncertainty about the location and orientation of the hole. Our approach in Chapter 5 is more generic in that a new insertion tasks can easily be learned via demonstration of a few insertions, and only a coarse estimate of the hole orientation is required at task execution time.

A recent work that also uses measured force to adapt the reference trajectory is (Abu-Dakka et al., 2014). In that work, an insertion is assumed to follow a certain position and force profile, which are learned from demonstration. The approach uses proportional feedback of the contact force and Iterative Learning Control respectively for online and long term adaptation of the reference position to achieve the desired contact force.

2.1.6 DYNAMICAL SYSTEMS: OVERVIEW AND HISTORICAL PERSPECTIVE

Dynamical Systems have emerged as one of the most general and flexible ways of representing motion plans for robotic applications. As reviewed below, numerous DS formulations with different properties have been introduced in the literature. What they all have in common is that motion is represented by a set of differential equations and hence technically allow to encode rich behavior including correct response to perturbations, generalization of a learned task etc.

Motion generation with dynamical systems is a long-standing research topic with important early approaches including the Vector Integration To End-point (VITE) model (Bullock and Grossberg, 1988) which was introduced to model human reaching motions. Recurrent Neural Networks (RNN) have been successfully used for modeling dynamics in various applications (Pearlmutter, 1989; Ito, 2004; Lin et al., 1995). However, neural network approaches typically suffer from long training times and more importantly difficulty to ensure basic desired properties such as stability or convergence to a limit cycle, which are the two most important topologies when using DS for robot motion representation.

A large body of work has been concerned with specifying a task-specific potential function and using its gradient as control (Khatib, 1986; Koditschek, 1988). The concept of navigation functions free from local minima builds on the same concept (Rimon and Koditschek, 1992). These approaches can be seen as DS approaches with a DS that is implicitly specified via a scalar potential function. It is straightforward to specify such functions for simple tasks without complex motion patterns, but these methods are limited in the range of motions that they can encode.

More recently, the Dynamic Motor Primitives (DMP) framework (Schaal, 2003; Ijspeert et al., 2002) and variants (Calinon et al., 2012) have gained popularity both in imitation learning (Schaal et al., 2003) and reinforcement learning

(Kober and Peters, 2010; Kormushev et al., 2010; Buchli et al., 2011). The DMP has essentially three components, 1) a canonical system yielding the evolution of the behavioral *phase variable* 2) a nonlinear forcing term, which is a function of the phase variable and 3) the transformation system, which is the sum of a simple linear attractor system and the nonlinear forcing term. The canonical system is defined as a linear system (first or second order) that converges to zero. The influence of the nonlinear forcing term is controlled by the phase variable and decays to zero along with the phase variable. Hence, after convergence of the phase variable, one is left with a simple linear system with well-known stability properties. Multiple dofs can be synchronized by sharing the canonical phase variable. DMPs have been shown to be an incredibly powerful tool for modeling both rhythmic and discrete motions. The learning in DMP consists in estimation of the nonlinear forcing term. This is usually done by linear regression on feature vectors that are predetermined kernel functions in the phase space, allowing fast and reliable learning even in high dimensions. Considering a state-vector including the controlled kinematic variable (e.g. joint positions) *and* the canonical phase, the DMP equations form an autonomous dynamical system. It is important to note that the kinematic variable is *not autonomous in isolation*. The DMP does not depend on time explicitly, however it does so implicitly because the phase variable evolves purely as a function of time. To deal with this issue, it has been suggested to introduce coupling from the controlled state back to the differential equation governing the evolution of the phase variable (Ijspeert et al., 2002). While such feedback coupling from the controlled state to the phase can be very useful in particular situations (Pastor et al., 2011), it is generally no longer possible to guarantee stability when such coupling is introduced. This is because the stability analysis of DMP relies on the phase decaying to zero, which is no longer guaranteed in the presence of coupling from the controlled state. In Chapter 7, we show that coupling on the form proposed in (Ijspeert et al., 2013) can result in problematic behavior unless carefully tuned. A detailed description of the DMP framework along with a recent review on developments in DMP research is given in, (Ijspeert et al., 2013).

Another class of DS representations define DS purely as a function of the controlled state, and are in contrast to DMP autonomous when considering as state vector the kinematic variable of interest (e.g. joint positions) (Gribovskaya et al., 2011). In recent years, significant research in our group² has been directed to the development of DS formulations of this type. The method suggested in (Gribovskaya et al., 2011) was based on using GMM for modeling demonstrations of position and velocity of a task. An autonomous DS is then obtained through GMR, considering the position as input and the velocity as output. A significant improvement was introduced with the Stable Estimator of Dynamical Systems (SEDS) in (Khansari-Zadeh and Billard, 2011), which adds

²The Learning Algorithms and Systems Laboratory, EPFL

constraints to the learning that ensures global asymptotic stability of the resulting DS. SEDS has since been adapted for hitting motions (Kronander et al., 2011), coordinated reach-grasp motions (Shukla and Billard, 2011) and fast reactive motion generation for catching flying objects (Kim et al., 2014). A notable difference to the DMP framework is that the nonlinearities are encoded in the position space and not in an auxiliary phase space as in DMP. This increases the robustness to perturbations, allows better generalization and furthermore gives the possibility to encode completely different behaviors in different parts of the state space (Gribovskaya et al., 2011). However, learning from demonstrations with a GMM and especially with SEDS is a considerably more difficult problem than one-shot least squares fit possible in DMP. GMM fitting is subject to local minima and relies on good initialization, which is not the case for DMP.

Extreme Learning Machines (ELM) (Huang et al., 2006) have been applied to learning DS from data in (Lemme et al., 2013). Stability is ensured locally by evaluating a quadratic Lyapunov function on a set of cleverly selected points in the workspace. Improvements including automatic choice of the most appropriate quadratic Lyapunov function candidate were recently published in (Lemme et al., 2014).

The power of using a DS for representing robot motion lies in the capability of encoding not only nominal motion from each starting point but also an appropriate response to perturbations. This is in stark contrast to the traditional way of modeling robot tasks which is typically a two step approach 1) plan a trajectory 2) execute trajectory with fixed controller. In this case, arbitrarily complex trajectories can be encoded in 1) but the response to perturbations is always the same and determined solely by 2). A branch of motion planning deals with this shortcoming by using feedback in the planning stage LaValle (2006) and continuously re-planning the motion during task execution, taking state measurements into account. While this can lead to increased robustness, most motion planning algorithms require significant time to re-plan a path, which means that these approaches can react to perturbations to some degree but not in the continuous manner that the DS approach does. In essence, this is due to the fact that the feedback motion planning approach optimizes an entire path while the DS approach requires simply the evaluation of a known function at each time step.

2.1.7 INCREMENTAL LEARNING IN DYNAMICAL SYSTEMS

Incremental learning from demonstration can alleviate the difficulty of simultaneously demonstrating desired behavior in multiple degrees of freedom. Furthermore, it can allow refinement and reuse of a learned model for a different task. Various methodologies have been used. In (Ogata et al., 2005), a neural network based approach inspired by how humans consolidate existing knowl-

edge is presented. Gaussian Mixture Modeling (GMM) usually in combination with Gaussian Mixture Regression (GMR) is a well-established framework in the RLfD community (Calinon et al., 2007). GMM are usually trained offline with the EM algorithm (Dempster et al., 1977), but incremental variants exist (Neal and Hinton, 1998; Song and Wang, 2005), allowing incremental RLfD based on GMM (Calinon and Billard, 2007; Cederborg et al., 2010). To deal with synchronization and clustering of motion data, Hidden Markov Models (HMM) have been extensively used. In (Kulic et al., 2008), a system based on HMM that incrementally learns whole body motion primitives from motion capture data. In (Lee and Ott, 2011), a specialized impedance control law designed to facilitate incremental kinesthetic teaching was used together with a HMM for representing the incoming data, allowing to elegantly rid the system of synchronization problems between demonstrations. Similarly to HMM, an autonomous DS representation does not have issues with synchronization, in this case because no temporal information is encoded in the model. In general, autonomous DS models seem particularly well-suited for incremental learning settings, but so far there has been little research in this direction.

The DMP framework with locally weighted learning in the phase space can use recursive least squares to incrementally update the parameters of the non-linear forcing term. However, DMP is designed to learn from a single demonstration, since it encodes one stereotypical motion (Ijspeert et al., 2013) as opposed to a family of possible trajectories like e.g. SEDS (Khansari-Zadeh and Billard, 2011) or ELM-based approaches (Lemme et al., 2014).

The optimization under constraints of SEDS (Khansari-Zadeh and Billard, 2011) makes it difficult to learn incrementally in this formulation. A more generally applicable method was proposed in (Khansari-Zadeh and Billard, 2014), which presents an approach that can stabilize any DS by online generation of an auxiliary command that ensures monotonic decay of a task-based energy function which is learned from demonstrations. This method allows more complex motions than stability constraints based on a quadratic energy function, which is used e.g. in (Khansari-Zadeh and Billard, 2011), but is still limited by the energy function used as a basis for the stabilization mechanism. Task-based Lyapunov functions have also been explored in the ELM framework in (Neumann et al., 2013; Lemme et al., 2014), although in that case on a more restricted quadratic form. All of these methods are based on using a parameterized Lyapunov function for ensuring asymptotic stability of the dynamics. In each case, this has consequences on the accuracy at which trajectories can be represented. In addition, these approaches only admit to learn incrementally insofar that the incrementally arriving data are consistent with the energy function, which is learned offline.

In chapter 6, we introduce a novel DS representation termed Locally Modulated Dynamical Systems (LMDS). This formulation is based on an existing DS which is locally transformed by training data, which can be arriving incremen-

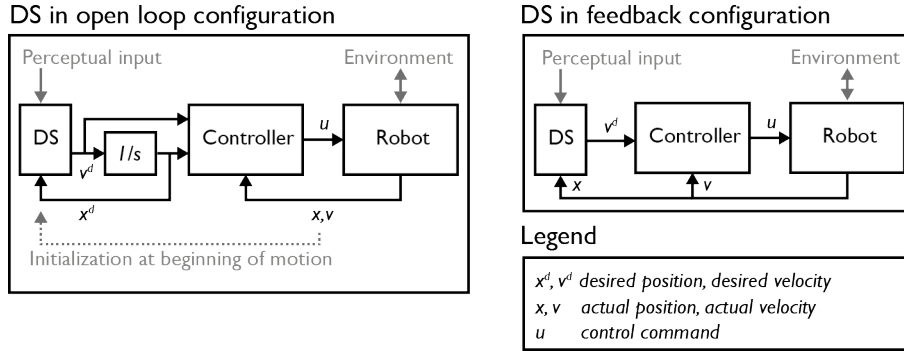


Figure 2.3: Illustration of open loop and closed loop control configurations with DS. In the open loop configuration, the DS is updated with the desired position resulting from integration of the desired velocity. The actual position of the robot is only used for initializing the integration at the beginning of the motion. In contrast, the feedback configuration continuously updates the DS with the actual position and realizes a control on the velocity error.

tally or in batches. The stability analysis is not based on a known Lyapunov function, and instead the formulation is 1) inherently incapable of introducing spurious attractors and 2) guaranteed to generate bounded trajectories. These are weaker properties than asymptotic stability, with the consequence that our dynamics can converge to limit cycles or orbits (but not spurious attractors). In exchange, we can directly incorporate incremental demonstrations, which need not comply with an energy function. As is shown in Chapter 6, asymptotic stability is for all practical purposes an unnecessary restriction in the LMDS framework, since it is not violated unless the demonstrations explicitly indicate orbital behavior.

2.1.8 CONTROL OF TASKS DESCRIBED BY DYNAMICAL SYSTEMS

The DS approaches reviewed in Sections 2.1.6 and 2.1.7 describe the *desired* behavior of the robot. The output of the DS task models during task execution is desired velocity or acceleration (which can be integrated to find a desired position). The most common architecture is to track the DS output using e.g. position controller³ (Kronander et al., 2011), parallel position/force control (Kalakrishnan et al., 2011) or impedance control (Buchli et al., 2011; Kronander and Billard, 2013). Unfortunately, in such a configuration, the possibility of the DS to intelligently respond to a perturbation is partially lost. While it can still respond appropriately to perceptive perturbations, such as a moving goal position or obstacle, it can not react to perturbations on the robot body because the DS is agnostic to the state of the robot, see Fig. 2.3.

Motivated by the need to develop control laws that render the robot passive w.r.t to external forces applied to it, Li and Horowitz introduced Passive Velocity

³The position trajectory can be integrated from the starting position of the motion.

Field Control (PVFC) in (Li and Horowitz, 1999). The authors highlighted the fact that time-dependent trajectory tracking controllers are generally not passive when considering the external forces and velocity of the robot as input-output pair. To remedy this issue, the authors advocated modeling of tasks in terms of velocity fields. No attention was given to the acquisition of such vector fields. As it happens, velocity field is just another word for what we refer to as DS in the preceding sections. But the approach of Li and Horowitz as well as related work by Duindam et al. has previously not been applied in the control of tasks learned and encoded by DS as reviewed in Sections 2.1.6 and 2.1.7.

The PVFC introduced in (Li and Horowitz, 1999) is strongly based around skew-symmetric velocity feedback terms, which have the effect of redirecting kinetic energy along the desired directions of the velocity field. A virtual fly-wheel is used to store energy internally in the system. It is shown that the robot under PVFC will converge to a multiple of the desired DS. It is hence possible to traverse the integral curves of the DS in positive or negative direction. This, as well as the speed of the robot, is determined by the initial conditions. An extension including adaptive parameter estimation was presented in (Li, 1999). Applications of PVFC include smart exercise machines (Li and Horowitz, 1997a,b) and rehabilitation devices (Erdogan et al., 2011; Erdogan and Patoglu, 2012).

A strongly related work to PVFC is (Duindam et al., 2004). In this case, the authors focus on the curve tracking problem and assumes the availability of a potential field with a valley along the desired curve. The tangent space of the desired curve is used to generalize desired directions of movement points that lie outside the desired curve. Similarly to PVFC, power-continuous controls are then applied for redirecting kinetic energy along the desired direction while convergence to the desired curve is achieved by potential energy shaping using the potential field associated with the desired curve. A coordinate-free derivation of the controller was also presented in (Duindam and Stramigioli, 2004). A notable contribution of this work is the possibility to *passively* compensate for the robot non-linearities along a desired direction of motion.

The controllers proposed in (Li and Horowitz, 1999) and (Duindam et al., 2004) have sound theoretical foundations and a number of interesting properties, the most important being passivity with respect to external force and robot velocity as input-output pair. This means that the robot will be stable in free motion but more importantly in contact with any passive environment (van der Schaft, 2000; Stramigioli, 2001). While these developments allow high performance tracking, neither gives much attention to the behavior of the system when interacting with an environment. On the other hand, the classical approach of a reference trajectory feeding an impedance controller yields a behavior in contact that is well understood. So far, we have discussed two possible routes for using DS in control: 1) the open-loop approach with impedance control and 2) closed-loop approach with a velocity field controller. The former allows us to

use existing intuition about appropriately setting control parameters for achieving desired interactive behavior, while the latter allows us to fully exploit the potential of a DS motion representation and in addition can guarantee passivity. Is it possible to combine these two approaches to get the best of both worlds? In chapter 7 we attempt to answer this question and propose a new controller which similar to (Li and Horowitz, 1999) and (Duindam et al., 2004) is passive but in contrast to these uses dissipation as an important part of the controller.

2.2 Technical preliminaries

This section briefly review some the most important technical tools that are used in this thesis. All material presented here is standard textbook material or previously published results. Sections 2.2.1 and 2.2.2 reviews DS and stability analysis. As a comprehensive introduction to this vast topic would require far more space than can be accommodated in this thesis, the material rather consists of cherry-picked definitions and stability theorems that are used later in the thesis. Sections 2.2.3 and 2.2.4 zoom in to a particular case of DS, namely dynamics of a robot manipulator and the various controllers that are used in the thesis. Lastly, Section 2.2.5 provides a short introduction to Gaussian Mixture Regression (GMR) and Gaussian Process Regression (GPR), two nonlinear regression algorithms that are used at several places in the thesis.

2.2.1 DYNAMICAL SYSTEMS

Dynamical Systems (DS) will be used in one form or another in every chapter of this thesis. Here, we summarize some definitions and properties related to this topic. The material presented in sections consists of cherry-picked items that can be found in any textbook on Dynamical Systems. For further details on this fascinating topic, the reader is referred (Slotine and Li, 1991) and (Khalil, 2002).

Let $\boldsymbol{\xi} \in \mathbb{R}^d$ denote a state vector of dimension $d \in \mathbb{N}$. Generally, $\boldsymbol{\xi}$ could be an arbitrarily rich description of a system. In this thesis, it will always represent the position and/or velocity of a robot manipulator in some coordinates. The DS that are relevant for this thesis are formulated as:

$$\dot{\boldsymbol{\xi}} = \mathbf{f}(t, \boldsymbol{\xi}, \mathbf{u}) \quad (2.1)$$

where t represents time and \mathbf{u} is an input vector of some dimensionality. The input vector u can be used to manipulate the evolution of the state of the system. In feedback control, the input is defined as a function of the state, $\mathbf{u} = \mathbf{u}(\boldsymbol{\xi})$. Hence, the *closed loop* dynamics becomes a function only of time and the state vector:

$$\dot{\boldsymbol{\xi}} = \mathbf{f}(t, \boldsymbol{\xi}, \mathbf{u}(\boldsymbol{\xi})) = \mathbf{f}'(t, \boldsymbol{\xi}) \quad (2.2)$$

DS of this form are often referred to as *unforced*, stemming from Eq. (2.1) with $u \equiv 0$. The term unforced is slightly misleading, since the same form can be used to describe a system that is actively controlled with feedback, as in Eq. (2.2).

A DS without dependency on any variables but the internal state is said to be *autonomous*. In this case, the evolution of the state vector can be written as:

$$\dot{\boldsymbol{\xi}} = \mathbf{f}(\boldsymbol{\xi}) \quad (2.3)$$

In this thesis, \mathbf{f} will always be a continuous, real valued function. Below, we list a set of important definitions related to the boundedness and equilibrium points of a DS:

Definition 2.1 (Boundedness). A DS on the form given by Eq. (2.2) is bounded if for each $\delta > 0$, there exists $\epsilon > 0$ such that:

$$\|\boldsymbol{\xi}(t_0)\| < \delta \Rightarrow \|\boldsymbol{\xi}(t)\| < \epsilon, \forall t > t_0$$

A DS may have points which attract or repulse system trajectories. Naturally, these are of great importance and are formally defined as follows:

Definition 2.2 (Equilibrium point). A point $\boldsymbol{\xi}^* \in \mathbb{R}^N$ such that $\mathbf{f}(\boldsymbol{\xi}^*, t) \equiv \mathbf{0} \forall t > t_0$ is an *equilibrium point* of \mathbf{f} .

We can now introduce the formal definition of *stability*⁴.

Definition 2.3 (Lyapunov stability). An equilibrium point $\boldsymbol{\xi}^*$ is said to be *stable in the sense of Lyapunov* or simply *stable* if for each $\epsilon > 0$ there exists $\delta = \delta(\epsilon, t_0) > 0$ such that:

$$\|\boldsymbol{\xi}(t_0) - \boldsymbol{\xi}^*\| < \delta \Rightarrow \|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^*\| < \epsilon, \quad \forall t > t_0$$

The interpretation of this definition is that an equilibrium point is stable, if and only if it is possible to remain arbitrarily close to it provided that the trajectory starts *close enough*. If trajectories in addition approach the equilibrium point over time, the equilibrium point is said to be *asymptotically stable*, defined below:

Definition 2.4 (Asymptotic stability). An equilibrium point $\boldsymbol{\xi}^*$ is called *asymptotically stable* if it is stable and, if in addition there exists $R(t_0) > 0$ such that:

$$\|\boldsymbol{\xi}(t_0) - \boldsymbol{\xi}^*\| < R \Rightarrow \|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^*\| \rightarrow 0, t \rightarrow \infty$$

If R can be arbitrarily large, the equilibrium point is *globally asymptotically stable*.

⁴Stability and the alternative term Lyapunov stability are used interchangeably in this thesis.

These definitions are valid for autonomous and non-autonomous DS. Note, that in the case of autonomous DS, the dependency of δ and R on t_0 disappear. Although Definitions 2.3 and 2.4 are properties associated to an equilibrium point, we will refer to a DS with a single equilibrium point that is stable as a stable DS.

For non-autonomous DS, the stability or instability of an equilibrium point may well depend on the time at which the trajectory started t_0 . In order to characterize stability that is independent on the initial time t_0 , the concept of uniform stability is used:

Definition 2.5 (Uniform (asymptotic) stability). An equilibrium point ξ^* is said to be *uniformly (asymptotically) stable* if it is (asymptotically) stable and if the scalar δ and R in Definitions 2.3 and 2.4 can be chosen independently of t_0 .

The stability definitions listed above deals with the behavior of an unforced DS in vicinity of its equilibria. They can be used to classify the behavior of a DS with *known* and modeled inputs, by first deriving the closed-loop system and then using the tools above.

If unknown inputs are present, for example if the system interacts with another system, the passivity formalism provides an extremely useful set of tools. Consider now a system with inputs $\mathbf{u} \in \mathbb{R}^M$ and outputs $\mathbf{y} \in \mathbb{R}^M$:

$$\dot{\xi} = \mathbf{f}(\xi) + \mathbf{g}(\mathbf{u}) \quad (2.4a)$$

$$\mathbf{y} = \mathbf{h}(\xi, \mathbf{u}) \quad (2.4b)$$

Passivity states that the system can never store more energy than is supplied to it.

Definition 2.6 (Passivity). A system on the form (2.4) is passive if there exists a lower bounded storage function $V : \mathbb{R}^N \mapsto \mathbb{R}_{\geq 0}$ such that:

$$V(\xi(t)) - V(\xi(0)) \leq \int_0^t \mathbf{u}(s)^T \mathbf{y}(s) ds$$

is satisfied for all $t \geq 0$, all input functions \mathbf{u} and all initial conditions $\xi(0) \in \mathbb{R}^N$.

With restriction to continuously differentiable storage functions (which are exclusively used in this thesis), an alternative definition which is often more convenient can be used:

Definition 2.7 (Alt. Passivity). A system on the form (2.4) is passive if there exists a continuously differentiable lower bounded storage function $V : \mathbb{R}^N \mapsto \mathbb{R}_{\geq 0}$ such that along the trajectories of (2.4):

$$\dot{V}(t) \leq \mathbf{u}(t)^T \mathbf{y}(t)$$

for all $t > 0$ and for all input functions $\mathbf{u}(t)$.

2.2.2 STABILITY ANALYSIS OF DYNAMICAL SYSTEMS

This section, like the preceding one, consists of standard textbook material. Further details, examples and explanations can be found in (Slotine and Li, 1991) and (Khalil, 2002).

A particularly pleasant class of DS are *linear* DS. In the unforced or closed-loop case, these take the form:

$$\dot{\boldsymbol{\xi}} = \mathbf{A}\boldsymbol{\xi} \quad (2.5)$$

where \mathbf{A} is a constant matrix. Note that this representation can be used without loss of generality for describing autonomous linear systems, since equilibrium points other than the origin can be achieved through a change of variables. The stability of a linear system is characterized by the eigenvalues of the matrix \mathbf{A} :

Theorem 2.1. *A system on the form given by Eq. (2.5) is globally asymptotically stable at the origin if and only if the real part of all eigenvalues $\lambda_i, i = 1 \dots d$ of \mathbf{A} are strictly negative:*

$$re(\lambda_i) < 0 \quad \forall i = 1 \dots d$$

For non-linear systems there exists no such generic method for determining stability. The century old results of Lyapunov's studies of DS still feature prominently in control theory and in nonlinear stability analysis in particular. In this thesis, we primarily make use of Lyapunov's direct method, which involves searching for a scalar function that can be used to characterize the stability properties of a particular nonlinear system.

Theorem 2.2 (Stability of equilibrium point in autonomous nonlinear DS). *Let $\boldsymbol{\xi} = \mathbf{0}$ be an equilibrium point of a dynamical system on the form $\dot{\boldsymbol{\xi}} = \mathbf{f}(\boldsymbol{\xi})$. Let $\mathcal{D} \subseteq \mathbb{R}^d$ be a region including the origin. Let $V(\boldsymbol{\xi})$ be a continuously differentiable function such that:*

(1) *V is positive definite in \mathcal{D} :*

$$V(\mathbf{0}) = 0 \quad \text{and} \quad V(\boldsymbol{\xi}) > 0 \quad \forall \boldsymbol{\xi} \in \mathcal{D} \setminus \mathbf{0}$$

(2) *\dot{V} is negative semidefinite in \mathcal{D} :*

$$\dot{V}(\boldsymbol{\xi}) \leq 0 \quad \forall \boldsymbol{\xi} \in \mathcal{D} \setminus \mathbf{0}$$

Then, $\boldsymbol{\xi} = \mathbf{0}$ is a stable. If, in addition:

(3) *\dot{V} is strictly negative definite in \mathcal{D} :*

$$\dot{V}(\boldsymbol{\xi}) < 0 \quad \forall \boldsymbol{\xi} \in \mathcal{D} \setminus \mathbf{0}$$

Then $\boldsymbol{\xi} = \mathbf{0}$ is locally asymptotically stable.

Note that Theorem 2.2 generalizes readily to equilibrium points located anywhere in \mathbb{R}^d through a change of variables. The local asymptotic stability property implies that any trajectory starting in \mathcal{D} will converge to the origin. The following important theorem of *global* asymptotic stability require some additional properties of V :

Theorem 2.3 (Global asymptotic stability of equilibrium point in autonomous nonlinear DS). *Let $\xi = \mathbf{0}$ be an equilibrium point for a system on the form $\dot{\xi} = \mathbf{f}(\xi)$. Let V be a continuously differentiable scalar function such that:*

(1) V is positive definite:

$$V(\mathbf{0}) = 0 \quad \text{and} \quad V(\xi) > 0 \quad \forall \xi \neq 0$$

(2) \dot{V} is negative definite:

$$\dot{V}(\xi) < 0 \quad \forall \xi \neq 0$$

(3) \dot{V} is radially unbounded:

$$\|\xi\| \rightarrow \infty \Rightarrow V(\xi) \rightarrow \infty$$

Then, $\xi = \mathbf{0}$ is globally asymptotically stable.

Slight variations are required for proving uniform stability in non-autonomous systems:

Theorem 2.4 (Stability of equilibrium point in non-autonomous nonlinear DS). *Let the origin be an equilibrium point for a system on the form given by Eq. (2.2). Let $V(\xi, t)$ be a continuously differentiable scalar function such that:*

(1) V is positive definite:

$$V(\xi, t) > 0 \quad \forall \xi \neq 0, \forall t \geq 0 \quad \text{and} \quad V(\mathbf{0}, t) = 0, \forall t \geq 0$$

(2) \dot{V} is negative semidefinite:

$$\dot{V}(\xi, t) \leq 0, \forall \xi \neq 0, \forall t \geq 0 \quad \text{and} \quad \dot{V}(\mathbf{0}, t) = 0, \forall t \geq 0$$

(3) V is decrescent:

$$\exists V'(\xi) > 0, \forall \xi \neq 0: \quad V(\xi, t) \leq V'(\xi), \forall t \geq 0$$

Then, the origin is uniformly stable. If, in addition:

(4) \dot{V} is negative definite:

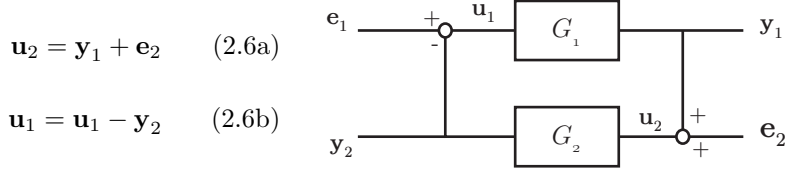
$$\dot{V}(\xi, t) < 0, \forall \xi \neq 0, \forall t \geq 0 \quad \text{and} \quad \dot{V}(\mathbf{0}, t) = 0, \forall t \geq 0$$

(5) V is radially unbounded in ξ :

$$\|\xi\| \rightarrow \infty \Rightarrow V(\xi, t) \rightarrow \infty$$

Then, the origin is globally uniformly asymptotically stable.

The *passivity theorem* characterizes interconnection between passive subsystems. Let G_1 and G_2 be two passive as Eq. (2.4b), satisfying passivity (Definition 2.6), connected in feedback configuration as follows:



Theorem 2.5. *Suppose G_1 and G_2 are passive. Then their feedback connection given by Eq. (2.6) with input $\mathbf{e} = [\mathbf{e}_1^T, \mathbf{e}_2^T]^T$ and output $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T]^T$ is also passive. In particular, if $\mathbf{e}_2 \equiv \mathbf{0}$, the feedback connection is passive with the input-output pair $\mathbf{e}_1, \mathbf{y}_1$.*

It should be noted that this is a simplified state-space version adapted to the needs of this thesis. For the more general passivity theorem and extensions to output-strict passivity the reader is referred to (van der Schaft, 2000).

2.2.3 RIGID BODY DYNAMICS

The dynamics of a robot manipulator with revolute joints can be modeled as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_c + \boldsymbol{\tau}_e \quad (2.7)$$

where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^N$ denote the position, velocity and acceleration of N joints of the robot, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{N \times N}$ the symmetric and positive definite joint space inertia matrix, $\mathbf{C}(\dot{\mathbf{q}}, \mathbf{q}) \in \mathbb{R}^{N \times N}$ the coriolis/centrifugal matrix and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^N$ the torque due to gravity. The control torques and external torques are denoted by $\boldsymbol{\tau}_c \in \mathbb{R}^N$ and $\boldsymbol{\tau}_e \in \mathbb{R}^N$ respectively.

The RBD can equivalently be described in Cartesian coordinates, using the manipulator Jacobian $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6 \times N}$ defined so that the Cartesian velocity corresponding to a joint velocity is given by $\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$. Let $\boldsymbol{\phi} \in \mathbb{R}^6$ denote a wrench vector acting on the end-effector of the robot. This wrench is distributed as torque in the joints of the robot according to the well-known relation $\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\boldsymbol{\phi}$. This relation can be used to implement a desired control wrench on the end-effector via the joint torques. The Cartesian RBD equations are written as:

$$\mathbf{M}_x(\mathbf{q})\ddot{\mathbf{x}} + \mathbf{C}_x(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{x}} + \mathbf{g}_x(\mathbf{q}) = \boldsymbol{\phi}_c + \boldsymbol{\phi}_e \quad (2.8)$$

where $\boldsymbol{\phi}_c, \boldsymbol{\phi}_e \in \mathbb{R}^6$ denote the control and external wrench respectively, $\mathbf{M}_x(\mathbf{q}) \in \mathbb{R}^{6 \times 6}$ and $\mathbf{C}_x(\dot{\mathbf{q}}, \mathbf{q})$ denote the Cartesian inertia and coriolis/centrifugal matrices respectively and where $\mathbf{g}_x(\mathbf{q}) \in \mathbb{R}^6$ denotes the Cartesian gravity vector. Note that the latter three *cannot* be described as functions solely of the variables $\dot{\mathbf{x}}, \mathbf{x}$, since a redundant robot has several possible joint configurations corresponding to the same Cartesian pose.

Consider an uncontrolled ($\boldsymbol{\tau}_c = 0$) manipulator. We can write the total energy of this system as the sum of the kinetic energy and the potential energy

due to gravity, denoted $V_g(\mathbf{q})$:

$$V(t) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} + V_g(\mathbf{q}) \quad (2.9)$$

Differentiating yields:

$$\dot{V}(t) = \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}} \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{M} \ddot{\mathbf{q}} + \frac{\partial V_g}{\partial \mathbf{q}} \dot{\mathbf{q}} \quad (2.10)$$

Substituting Eq. (2.7) and $\frac{\partial V_g}{\partial \mathbf{q}} = \mathbf{g}(\mathbf{q})$ yields:

$$\dot{V}(t) = \frac{1}{2} \dot{\mathbf{q}}^T (\dot{\mathbf{M}} - 2\mathbf{C}) \dot{\mathbf{q}} + \boldsymbol{\tau}_e^T \dot{\mathbf{q}} \quad (2.11)$$

The first term is identical to zero due to the skew-symmetry property of $\dot{\mathbf{M}} - 2\mathbf{C}$, yielding:

$$\dot{V}(t) = \boldsymbol{\tau}_e^T \dot{\mathbf{q}} \quad (2.12)$$

which proves passivity of the uncontrolled manipulator. This result was first exploited by the seminal paper (Takegaki and Arimoto, 1981) to prove stable regulation at a set point by simple modification of the manipulators potential energy.

2.2.4 IMPEDANCE CONTROL

Although most applications seem to focus on Cartesian impedance control, an impedance can be defined in any choice of coordinates. Let $\boldsymbol{\xi} \in \mathbb{R}^d$ represent the position of the robot in a suitable coordinate system (e.g. Cartesian space or joint space). Let $\boldsymbol{\xi}^r$ denote a *virtual* target configuration for the robot. Usually⁵, the goal of impedance control is to control the robot such that external generalized forces $\mathbf{F} \in \mathbb{R}^d$ interact with it according to the following dynamics:

$$\mathbf{H} \ddot{\boldsymbol{\xi}} + \mathbf{D} \dot{\boldsymbol{\xi}} + \mathbf{K} \tilde{\boldsymbol{\xi}} = \mathbf{F} \quad (2.13)$$

where $\mathbf{H}, \mathbf{D}, \mathbf{K} \in \mathbb{R}^{d \times d}$ are the target inertia⁶, damping and stiffness matrices respectively. As detailed below, the simplest implementations of impedance control restrict the choices of these parameters, notably the inertia matrix becomes a function of the configuration of the robot. The matrices $\mathbf{H}, \mathbf{D}, \mathbf{K}$ are written without arguments here, but they may be functions of the robot state, time or variables depending on 1) the choice of control implementation which restricts the choice of parameters and 2) task-based design, which may require or benefit from varying the impedance over time or as a function of some other variable.

⁵Impedance control is not limited to this second order system, but it is the most common form and there is good reason for it — it can be implemented with limited model knowledge of the robot.

⁶Strictly speaking, this system does not have a physical equivalent because inertias are not nodic elements, however this form is commonly used in robotics via inverse dynamics control.

The control algorithm presented here predates the whole concept of impedance control and represented a major breakthrough when it was published as it was one of the first position controllers with a rigorous stability analysis (Takegaki and Arimoto, 1981). Nonetheless, I choose to categorize it as an impedance control algorithm as the closed-loop behavior is a special case of Eq. (2.13).

Let $\mathbf{q}, \mathbf{q}^r \in \mathbb{R}^N$ denote the actual and desired configuration for an N-joint robot manipulator. Let $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}^r$. Consider the control vector given by:

$$\boldsymbol{\tau}_c = \mathbf{g}(\mathbf{q}) - \mathbf{K}\tilde{\mathbf{q}} - \mathbf{D}\dot{\tilde{\mathbf{q}}}$$
 (2.14)

where feedforward compensation of gravity is used with $\mathbf{g}(\mathbf{q})$ and where feedback is used to proportionally act on the configuration error with the symmetric positive definite stiffness matrix \mathbf{K} . In addition, velocities are damped through feedback of velocity acted upon by the positive definite damping matrix \mathbf{D} . Substitution of Eq. (2.14) in Eq. (2.7) yields the closed loop behavior:

$$\mathbf{M}(\mathbf{q})\ddot{\tilde{\mathbf{q}}} + \overline{\mathbf{D}}(\mathbf{q}, \dot{\tilde{\mathbf{q}}})\dot{\tilde{\mathbf{q}}} + \mathbf{K}\tilde{\mathbf{q}} = \boldsymbol{\tau}_e$$
 (2.15)

where $\overline{\mathbf{D}}(\mathbf{q}, \dot{\tilde{\mathbf{q}}}) = \mathbf{D} + \mathbf{C}(\mathbf{q}, \dot{\tilde{\mathbf{q}}})$ had been introduced. Equation (2.15) corresponds to Eq. (2.13) with a stationary virtual point, with a target inertia equal to the inherent configuration-dependent inertia and with a damping term that is nonlinear due to the influence of coupled inertial effects. This is the simplest possible implementation of impedance control and is surprisingly effective, especially in quasi-static cases. Note that strictly speaking, this controller is only valid for regulation. However, in practice it can work very well with a *slowly* moving reference configuration.

Now consider a more general scenario where the virtual trajectory is not a fixed point, but is defined at each time instance by vectors $\ddot{\mathbf{q}}^r, \dot{\mathbf{q}}^r, \mathbf{q}^r \in \mathbb{R}^N$. Let $\boldsymbol{\tau}_c$ be an inverse dynamics command:

$$\boldsymbol{\tau}_c = \mathbf{M}(\mathbf{q})\boldsymbol{\nu} + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) - \boldsymbol{\tau}_e$$
 (2.16)

Define the new control input $\boldsymbol{\nu}$ as:

$$\boldsymbol{\nu} = \ddot{\mathbf{q}}_t^r + \mathbf{H}^{-1}(-\mathbf{D}\dot{\tilde{\mathbf{q}}} - \mathbf{K}\tilde{\mathbf{q}} + \boldsymbol{\tau}_e)$$
 (2.17)

Substituting Equations (2.16) and (2.17) in Eq. (2.7) yields the closed loop dynamics:

$$\mathbf{H}\ddot{\tilde{\mathbf{q}}} + \mathbf{D}\dot{\tilde{\mathbf{q}}} + \mathbf{K}\tilde{\mathbf{q}} = \boldsymbol{\tau}_e$$
 (2.18)

which corresponds exactly to the generic impedance model in Eq. (2.13) without

any constraints on the impedance parameters or virtual trajectory.

SIMPLE CARTESIAN IMPEDANCE CONTROL

The name 'simple' impedance control was coined in (Hogan, 1985a), and referred to a controller with properties like the joint PD regulator described previously, but defined in a Cartesian work space. The version used in this thesis is a slightly refined version which improves upon the original implementation through the use of more robust, and physically consistent orientation error representation using axis/angle representation rather than the more commonly used Euler angles (Khatib, 1987) which suffers from representational singularities. The controller was originally proposed in (Caccavale et al., 1999).

Consider a controller given by:

$$\phi_c = \mathbf{g}_x(\mathbf{q}) - \mathbf{D}\dot{\mathbf{x}} - \mathbf{K}(\mathbf{x} - \mathbf{x}^r) \quad (2.19)$$

Substitution in the Cartesian RBD Equations yield:

$$\mathbf{M}_x(\mathbf{q})\ddot{\mathbf{x}} + \bar{\mathbf{D}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{x}} + \mathbf{K}\tilde{\mathbf{x}} = \phi_e \quad (2.20)$$

with $\bar{\mathbf{D}}(\dot{\mathbf{q}}, \mathbf{q}) = \mathbf{C}_x(\dot{\mathbf{q}}, \mathbf{q}) + \mathbf{D}$. This is a *regulator*, and can strictly speaking only be used with a fixed desired reference point. The Cartesian error representation $\tilde{\mathbf{x}} \in \mathbb{R}^6$ is discussed separately below.

CARTESIAN IMPEDANCE CONTROL

The 'real' Cartesian impedance controller that allows free specification of desired inertia, damping and stiffness is described here. Similarly to the joint space case, it requires force feedback and model knowledge for implementing the necessary inverse dynamics command:

$$\phi_c = \mathbf{M}_x(\mathbf{q})\nu_x + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{x}} + \mathbf{g}_x(\mathbf{q}) - \phi_e \quad (2.21)$$

with new control input $\nu_x \in \mathbb{R}^6$ defined as:

$$\nu_x = \ddot{\mathbf{x}}^r + \mathbf{H}^{-1}(-\mathbf{D}\dot{\tilde{\mathbf{x}}} - \mathbf{K}\tilde{\mathbf{x}} + \phi_e) \quad (2.22)$$

Substitution in Eq. (2.8) yields the following closed loop dynamics:

$$\mathbf{H}\ddot{\tilde{\mathbf{x}}} + \mathbf{D}\dot{\tilde{\mathbf{x}}} + \mathbf{K}\tilde{\mathbf{x}} = \phi_e \quad (2.23)$$

which correspond exactly to the target impedance model in Eq. (2.13) replacing ξ for \mathbf{x} .

CARTESIAN ERROR REPRESENTATION

The Cartesian controllers described above have in common that they rely on

the Cartesian error signal denoted as $\tilde{\mathbf{x}} \in \mathbb{R}^6$. It consists of two parts:

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{\mathbf{p}} \\ \tilde{\boldsymbol{\eta}} \end{bmatrix} \quad (2.24)$$

with $\tilde{\mathbf{p}} \in \mathbb{R}^3$ is simply the translational error computed as $\tilde{\mathbf{p}} = \mathbf{p} - \mathbf{p}^r$, where $\mathbf{p}, \mathbf{p}^r \in \mathbb{R}^3$ have been introduced to denote the actual and reference position of the end-effector respectively. The representation of the orientation error has historically been a topic of some debate. Despite the well-known problems of singularities, a minimal representation using Euler angles is often used for representing the error. This can be done in two ways. The traditional way is to first compute an Euler angle representations \mathbf{e}, \mathbf{e}^r corresponding to the actual and reference orientation $\mathbf{R}, \mathbf{R}^r \in \mathbb{R}^{3 \times 3}$ respectively, and then simply setting $\tilde{\boldsymbol{\eta}} = \mathbf{e} - \mathbf{e}^r$ (Khatib, 1987). A more robust way of using Euler angles is to first compute the orientation difference $\hat{\mathbf{R}} = \mathbf{R}^T \mathbf{R}^r$ and then extract the corresponding Euler angles $\hat{\mathbf{e}}$, and using this as the orientation error $\tilde{\boldsymbol{\eta}} = \hat{\mathbf{e}}$ (Caccavale et al., 1999). The latter avoids singularities for small errors and can hence be used successfully in applications with high rotational impedance that hinders significant orientation errors to occur (Ott, 2008). Remarkably, Euler angles are still commonly used for error representation despite well known problems and alternative formulations Caccavale et al. (1999); Stramigioli (2001)

In this thesis, the orientation error is always computed using the angle/axis representation. This has several advantages, the two most important being 1) No singularities and 2) physical consistency between the orientation error and the torque part of the control wrench (Caccavale et al., 1999). This technique can easily be used with the reference orientation specified either as a unit quaternion or a full rotation matrix. For the latter case, the rotation difference $\hat{\mathbf{R}} = \mathbf{R}^T \mathbf{R}^r$ is first computed, and the corresponding angle/axis representation $\hat{\boldsymbol{\zeta}} \in \mathbb{R}^3$ is extracted from $\hat{\mathbf{R}}$. This then used as orientation error in Eq. (2.24), setting $\tilde{\boldsymbol{\eta}} = \hat{\boldsymbol{\zeta}}$. The procedure is analogous for the quaternion case, using quaternion algebra to compute the orientation difference and then computing the angle/axis representation of the resulting quaternion.

DESIGN OF THE IMPEDANCE PARAMETERS

The behavior of an impedance controller in contact can change dramatically depending on the structure of the impedance parameters. A 6×6 Cartesian stiffness matrix \mathbf{K} can be decomposed as:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_P & \mathbf{K}_{PR} \\ \mathbf{K}_{RP} & \mathbf{K}_R \end{bmatrix} \quad (2.25)$$

with $\mathbf{K}_P \in \mathbb{R}^{3 \times 3}$ relating forces to positional errors, $\mathbf{K}_{PR} \in \mathbb{R}^{3 \times 3}$ relating forces to rotational errors, $\mathbf{K}_{RP} \in \mathbb{R}^{3 \times 3}$ relating torques to positional errors and $\mathbf{K}_R \in \mathbb{R}^{3 \times 3}$ relating torques to rotational errors. In this thesis, the Carte-

sian controllers that are used impose a certain structure so that translational errors only result in force response and rotational errors only result in torque response. This is achieved by choosing $\mathbf{K}_{PR} = \mathbf{K}_{RP} = \mathbf{0}$. This decomposition is physically possible close to the equilibrium of the stiffness term (Loncaric, 1987; Zefran and Kumar, 2002). Furthermore, the rotational part of the stiffness is assigned a diagonal structure $\mathbf{K}_R = \text{diag}([k_R^x, k_R^y, k_R^z]^T)$, where the elements on the diagonal represent the rotational stiffness around the three axes of the end-effector frame of reference. In this thesis, the translational stiffness \mathbf{K}_p will often be allowed to vary, while \mathbf{K}_R will always be constant. The damping matrix will always be designed so as to achieve a constant damping ratio of the system:

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_R \end{bmatrix} \quad (2.26a)$$

Consider the eigen decomposition of the symmetric and positive definite translational stiffness⁷ $\mathbf{K}_P = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$. The translational damping is constructed around the same eigenvectors, and with eigenvalues that are proportional with constant $\nu > 0$ to the square-root of the corresponding eigenvalues of \mathbf{K}_P :

$$\mathbf{D}_P = \mathbf{Q}(\nu\mathbf{\Lambda}^{\frac{1}{2}})\mathbf{Q}^T \quad (2.26b)$$

The rotational damping is designed similarly as a function of the rotational damping, resulting in a diagonal rotational damping:

$$\mathbf{D}_R = \nu\mathbf{K}_R^{\frac{1}{2}} \quad (2.26c)$$

The constant ν can be tuned to adjust the damping of the system. In this thesis, the damping will always be designed according to Eq. (2.26) with $\nu = 2$ unless otherwise stated.

2.2.5 NONLINEAR REGRESSION

In several places, this thesis makes use of nonlinear regression techniques for various purposes. Especially, Gaussian Mixture Regression (GMR) and Gaussian Process Regression (GPR) are used extensively in the thesis. Below I briefly describe these techniques.

GAUSSIAN MIXTURE REGRESSION

The Gaussian Mixture Model (GMM) is a powerful tool with countless applications in signal processing, speech processing, nonlinear regression and pattern recognition etc. While a single Gaussian function is often too limited to describe complex distributions, a combination of Gaussian distributions is much

⁷ $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$ contains the orthonormal eigenvectors of the stiffness matrix in its columns, and $\mathbf{\Lambda} \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix that contains the eigenvalues on its diagonal.

more powerful. A GMM with K modeling the joint probability distribution of a multi-dimensional variable $\boldsymbol{\xi} \in \mathbb{R}^M$ has probability density function given by:

$$p(\boldsymbol{\xi}) = \sum_{k=1}^K \pi^k \mathcal{N}(\boldsymbol{\xi}; \mathbf{m}^k, \mathbf{C}^k) \quad (2.27)$$

where $\pi^k > 0$, $\mathbf{m}^k \in \mathbb{R}^M$, $\mathbf{C}^k \in \mathbb{R}^{M \times M}$ for $k = 1 \dots K$ are the priors, means and covariances of the Gaussians in the GMM. If $\boldsymbol{\xi}$ is partially observed, i.e. some entries of $\boldsymbol{\xi}$ are known whereas the remaining are unknown, we can condition Eq. (2.27) on the part of $\boldsymbol{\xi}$ that is observed. Therefore, let $\boldsymbol{\xi} = [\boldsymbol{\xi}_I, \boldsymbol{\xi}_O]^T$, where $\boldsymbol{\xi}_I$ denotes an ‘‘input’’, i.e. the part of $\boldsymbol{\xi}$ that is observable and $\boldsymbol{\xi}_O$ is the ‘‘output’’, i.e. the part of $\boldsymbol{\xi}$ whose p.d.f. we wish to find. The distribution of the output conditioned on the input can then be written:

$$p(\boldsymbol{\xi}_O | \boldsymbol{\xi}_I) = \sum_{k=1}^K p(k | \boldsymbol{\xi}_I) p^k(\boldsymbol{\xi}_O | \boldsymbol{\xi}_I) \quad (2.28)$$

where

$$p^k(\boldsymbol{\xi}_O | \boldsymbol{\xi}_I) = \mathcal{N}(\boldsymbol{\xi}_O | \boldsymbol{\xi}_I; \mathbf{m}_{O|I}^k, \mathbf{C}_{O|I}^k) \quad (2.29a)$$

$$\mathbf{m}_{O|I}^k = \mathbf{m}_O^k + \mathbf{C}_{OI}^k (\mathbf{C}_I^k)^{-1} (\boldsymbol{\xi}_I - \mathbf{m}_I^k) \quad (2.29b)$$

$$\mathbf{C}_{O|I}^k = \mathbf{C}_O^k - \mathbf{C}_{OI}^k (\mathbf{C}_I^k)^{-1} \mathbf{C}_{IO}^k \quad (2.29c)$$

with

$$\mathbf{m}^k = \begin{bmatrix} \mathbf{m}_I^k \\ \mathbf{m}_O^k \end{bmatrix}, \quad \mathbf{C}^k = \begin{bmatrix} \mathbf{C}_I^k & \mathbf{C}_{IO}^k \\ \mathbf{C}_{OI}^k & \mathbf{C}_O^k \end{bmatrix} \quad (2.29d)$$

The probability of Gaussian k being responsible for $\boldsymbol{\xi}$ given $\boldsymbol{\xi}_I$ is usually defined as:

$$p(k | \boldsymbol{\xi}_I) = \frac{\pi^k \mathcal{N}(\boldsymbol{\xi}_I; \mathbf{m}_I^k, \mathbf{C}_I^k)}{\sum_{i=1}^K \pi^i \mathcal{N}(\boldsymbol{\xi}_I; \mathbf{m}_I^i, \mathbf{C}_I^i)} \quad (2.30)$$

GMR is a functional relationship between the input $\boldsymbol{\xi}_I$ and the output $\boldsymbol{\xi}_O$, which is achieved by taking the mean of the conditional distribution in Eq. (2.28):

$$E\{\boldsymbol{\xi}_O | \boldsymbol{\xi}_I\} = \sum_{k=1}^K p(k | \boldsymbol{\xi}_I) \mathbf{m}_{O|I}^k \quad (2.31)$$

GMR is a convex sum of linear models, weighted non-linearly across the input space. The normalization of the mixing weights in Eq. (2.30) means that at least one of the linear models will always have a significant contribution to the regression signal. The practical effect of this normalization is generalization, i.e. that the relationships found locally in training data are used in inferring outputs for inputs which lie far from the training data in input space. This is often a desirable effect, since it can reduce the amount of training data needed for good inference. However, sometimes it is inappropriate to generalize in this manner. In appendix A.1, we propose a novel regression strategy for GMM,

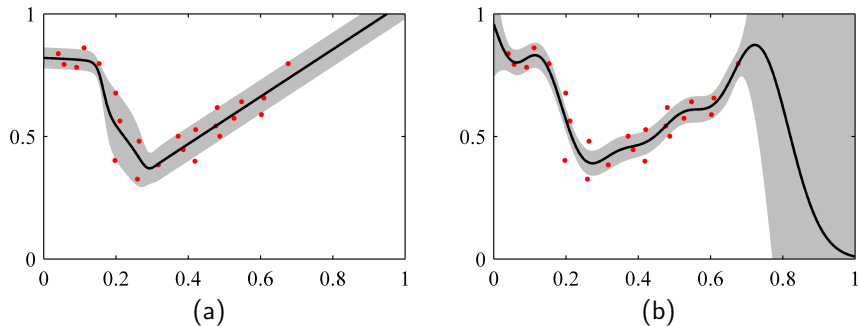


Figure 2.4: The figure shows regression with unidimensional input and output on a toy data set with GMR and GPR. The regression line is the central line and the enveloping lines indicates a confidence envelope corresponding to two standard deviations. **Left:** GMM with three components, fitted using the EM algorithm. **Right:** GPR with a squared exponential covariance function. Note that in contrast to GMR there is limited extrapolation outside the regions where training data is provided.

which preserves the advantages of the standard GMM/GMR formulation while it allows to control the degree of generalization with a single scalar parameter. An example of standard GMR on toy data is given in Fig. 2.4.

GAUSSIAN PROCESS REGRESSION

Let $\xi_{\mathcal{O}} \in \mathbb{R}^L$ be a vector containing L scalar observations of some process. Let $\xi_{\mathcal{I}}^1, \dots, \xi_{\mathcal{I}}^L \in \mathbb{R}^D$ be the associated D -dimensional input locations. In GPR, the goal is to make predictions of an unknown function f , which is assumed to underly the observed data. It is assumed that the observed outputs are noisy samples of this function:

$$\xi_{\mathcal{O}}^l = f(\xi_{\mathcal{I}}^l) + \epsilon^l, \quad l = 1 \dots L$$

where ϵ is i.i.d Gaussian noise with variance σ_n^2 . The joint distribution of the L training points, $\xi_{\mathcal{O}} = [\xi_{\mathcal{O}}^1, \dots, \xi_{\mathcal{O}}^L]^T$ and the output $\xi_{\mathcal{O}}^*$ at some query point $\xi_{\mathcal{I}}^*$ is then fully determined by a covariance function $k(\cdot, \cdot)$:

$$\begin{bmatrix} \xi_{\mathcal{O}} \\ \xi_{\mathcal{O}}^* \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma_n^2 \mathbf{I} & \mathbf{k}^* \\ \mathbf{k}^{*T} & k(\xi_{\mathcal{I}}^*, \xi_{\mathcal{I}}^*) \end{bmatrix} \right)$$

where

$$\mathbf{k}^* = [k(\xi_{\mathcal{I}}^1, \xi_{\mathcal{I}}^*), \dots, k(\xi_{\mathcal{I}}^L, \xi_{\mathcal{I}}^*)]^T$$

and where \mathbf{K} is a $L \times L$ matrix whose element at row i , column j is given by:

$$[\mathbf{K}]_{ij} = k(\xi_{\mathcal{I}}^i, \xi_{\mathcal{I}}^j)$$

Predictions are made by conditioning the joint distribution over training points and the query point:

$$\xi_{\mathcal{O}}^* | \xi_{\mathcal{O}} \sim \mathcal{N} \left(\mu_{\xi_{\mathcal{O}}^* | \xi_{\mathcal{O}}}, \sigma_{\xi_{\mathcal{O}}^* | \xi_{\mathcal{O}}}^2 \right)$$

The mean of the resulting distribution is used as estimator:

$$\mu_{\xi_{\mathcal{O}}^* | \xi_{\mathcal{O}}} = \mathbf{k}^{*T} [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \xi_{\mathcal{O}} \quad (2.33a)$$

with predictive variance:

$$\sigma_{\xi_{\mathcal{O}}^* | \xi_{\mathcal{O}}}^2 = k(\xi_{\mathcal{I}}^*, \xi_{\mathcal{I}}^*) - \mathbf{k}^{*T} [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{k}^* \quad (2.33b)$$

Unlike GMR, GPR supports only scalar outputs. If multiple variables are to be predicted this is done with a separate GP per output (although they can, of course, share the input locations). The performance of GPR as estimator is largely determined by the choice of covariance-function k . In this thesis, we use the standard squared exponential covariance function:

$$k(\xi_{\mathcal{I}}, \xi'_{\mathcal{I}}) = \sigma_s \exp \left(\frac{-1}{l^2} \|\xi_{\mathcal{I}} - \xi'_{\mathcal{I}}\|^2 \right) \quad (2.34)$$

where $\sigma_s > 0$ is signal variance scaling parameter $l > 0$ is the lengthscale determining tradeoff between detail and generalization. An example of GPR is shown in Fig. 2.4.

INTERACTION INTERFACE FOR COMPLIANT SKILL TRANSFER

3.1 Introduction

As reviewed in Section 2.1.1, the specification of task-based impedance profiles is a difficulty when using impedance control and varying impedance control in particular. Hand-tuning impedance profiles is tedious and requires expert knowledge. On the other hand, LfD has emerged as an incredibly efficient means of intuitively transferring skills to robots (refer to Section 2.1.3). An important question, hence, is how to incorporate varying impedance control into LfD. Basically, two components are needed: 1) A way of intuitively and efficiently transferring the desired impedance variations to the robot and 2) The possibility of augmenting the task model with impedance information. While the second of these problems is rather straightforward, the first represents a major challenge in LfD for compliant manipulation. This chapter introduces a novel human-robot interface that allows a human to teach a robot how it should vary its stiffness during the task.

As the motivation of LfD is to make it easy for users without knowledge of programming to teach tasks to their robots, any interface which is used in the teaching process for LfD should be intuitive. With this in mind, the developed teaching interface for variable stiffness is inspired by the way humans convey such information between each other. In dance and other sports, when the teacher wishes to convey to the student that she should relax, she may say 'relax' and at the same time wiggle the limb that is too stiff. Similarly, if the teacher wants to insist on the importance of a particular posture, this may be indicated by physically guiding the arm of the student into the desired posture and indicating the importance by a firm grip on the students arm. Our mechanism for decreasing and increasing the stiffness are based on these ideas. The robot is initially set to move along a desired trajectory with a default stiffness. The teacher intervenes along the trajectory to decrease or increase stiffness when needed. This is done by wiggling the robot's end effector around its equilibrium position, and following the robots movement with a firm grasp pressure, which is registered using an artificial skin module mounted on the robot. These interactions are illustrated in Fig. 3.1. The stiffness is updated online, so that the teacher can immediately feel the effect of the interaction. This is very important, since it is the only way that the teacher can know if the desired behavior

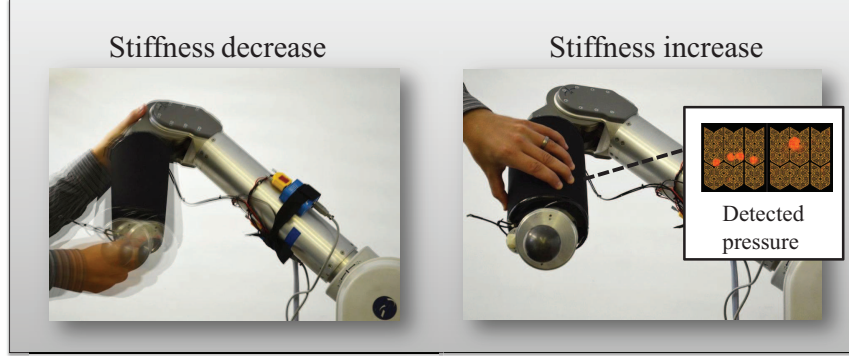


Figure 3.1: The figure shows the principles behind the proposed interface. The teacher interacts with the robot as it is moving to alter its stiffness. To decrease the stiffness the teacher wiggles the robot around its current position (left figure). To increase the stiffness, the teacher increases the grip force with which he holds the robot (right figure). The robot responds online to these stimuli, so that the teacher gets immediate haptic feedback on the new stiffness.

has been learned by the robot. The varying impedance profile can be associated to the task either temporally or spatially, and are naturally synchronized with the trajectory since the demonstrations are given during execution of the latter.

This chapter is organized as follows. Section 3.2 describes the Online Learning of Varying Stiffness (OLVS) algorithm which is the initial version of our interaction interface. It is limited to locally decreasing the singular values of the stiffness matrix and works with a time-indexed task description. This is then extended in Section 3.3 to include the grasp pressure modality for increasing the stiffness, as well as several other improvements. These sections correspond to the publications (Kronander and Billard, 2012) and (Kronander and Billard, 2013) respectively. A literature review on the topic treated in this chapter is given in Section 2.1.4.

3.2 Online Learning of Varying Stiffness

There exists already a large body of work for dealing with task transfer of kinematic tasks. Here, we hence assume the availability of a Cartesian reference trajectory that describes a task, and focus on how the user can augment this trajectory with varying impedance information. Our proposed solution is based on having the robot move along its trajectory using a high default stiffness. In the parts of the task where the teacher wants to reduce or change the principal axes of the stiffness matrix, he/she intervenes by wiggling the robot end-effector around its path. Note that the interface is concerned with adaptation of the translational stiffness component (refer to Section 2.2.4). The rotational components of the stiffness matrix are kept constant at a high value.

3.2.1 STIFFNESS ADJUSTMENT BASED ON INTERACTIONS

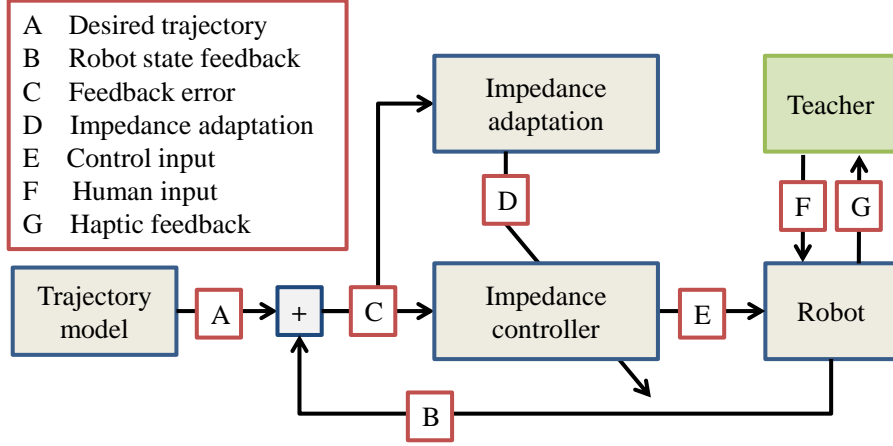


Figure 3.2: Overview of the interface. The stiffness is updated by observing the position deviations from the desired reference point due to the teachers interactions. The robot adapts its stiffness online, so the teacher gets direct haptic feedback of the effect of his interaction.

The way that the robot interprets spatial perturbations for adjusting shares similarities with how variability of a demonstration data set is used to set stiffness in Calinon et al. (2010). The basic idea is that if the teacher imposes perturbations with high variance in a direction, the robot should reduce its stiffness in that direction. The symmetric and positive definite stiffness matrix is built around the principal directions of the covariance matrix of the perturbation data, with stiffness along each direction set to be inversely proportional to the square root of the corresponding singular value of the covariance of the perturbations.

Let $\mathbf{x}, \mathbf{x}^d \in \mathbb{R}^3$ denote the position of the end-effector of the robot in a Cartesian work-space. We introduce the notation $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}^d$ for representing a perturbation data point. Let $\Xi = \{\tilde{\mathbf{x}}_j, t_j\}_{j=0}^J$ denote the set of observed perturbations with their corresponding time stamps, where J is the number of provided perturbation data. At time t , a stiffness matrix is assigned based on the data in Ξ with time stamps in the range $[t - S, t]$. Thus, the stiffness assignment is based on a sliding temporal window-view of length S over the observed perturbation data. Let L_t and U_t define the lower and upper bounds for the indices of data points inside the temporal window:

$$L_t = \max\{j \in [1, 2 \dots J] : t_j < t - S\} \quad (3.1a)$$

$$U_t = \min\{j \in [1, 2 \dots J] : t \leq t_j\} \quad (3.1b)$$

We denote by $N_t = U_t - L_t$ the number of data points in the sliding window at time t . Let \mathbf{M}_t^1 and \mathbf{M}_t^2 denote the first and second empirical moments of the

spatial data in the window:

$$\mathbf{M}_t^1 = \frac{1}{N_t} \sum_{j=L_t}^{U_t} \tilde{\mathbf{x}}_j \quad (3.2a)$$

$$\mathbf{M}_t^2 = \frac{1}{N_t} \sum_{j=L_t}^{U_t} \tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_j^T \quad (3.2b)$$

then, the covariance matrix is given by:

$$\boldsymbol{\Sigma}_t = \mathbf{M}_t^2 - \mathbf{M}_t^1 (\mathbf{M}_t^1)^T \quad (3.2c)$$

This covariance matrix is what determines the stiffness commanded to the robot at time t . The covariance matrix is symmetric and positive definite, so it can be decomposed as $\boldsymbol{\Sigma}_t = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T$, where $\boldsymbol{\Lambda}$ is a diagonal matrix composed of the singular values $\lambda_t^i > 0, i = 1, 2, 3$, and \mathbf{Q} is a matrix containing the orthonormal principal directions in its columns. The standard deviation of the data along each direction is given by $\sigma_t^i = \sqrt{\lambda_t^i}, i = 1, 2, 3$. The stiffness matrix \mathbf{K}_t is constructed using the same principal directions as the covariance matrix :

$$\mathbf{K}_t = \mathbf{Q}\boldsymbol{\Gamma}\mathbf{Q}^T \quad (3.3a)$$

with

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma(\sigma_t^1) & 0 & 0 \\ 0 & \gamma(\sigma_t^2) & 0 \\ 0 & 0 & \gamma(\sigma_t^3) \end{bmatrix} \quad (3.3b)$$

where the singular values are set negatively proportional to the square root of the corresponding singular value of the covariance matrix:

$$\gamma(\sigma^i) = \begin{cases} \underline{k} & \bar{\sigma} < \sigma_t^i \\ \bar{k} - (\bar{k} - \underline{k}) \frac{\sigma_t^i - \underline{\sigma}}{\bar{\sigma} - \underline{\sigma}} & \underline{\sigma} \leq \sigma_t^i \leq \bar{\sigma} \\ \bar{k} & \sigma_t^i < \underline{\sigma} \end{cases} \quad (3.3c)$$

for $i = 1, 2, 3$. The admissible values for the stiffness in any direction is bounded below by \underline{k} and above by \bar{k} . These, along with the $\underline{\sigma}$ and $\bar{\sigma}$ are tunable parameters of the system.

The teaching process consists of perturbing the robot while it is performing task, i.e. providing a stream of data points which are added to $\boldsymbol{\Xi}$. The data set is sorted in order by increasing time, and new data is simply inserted in the place corresponding to the time at which the perturbation was perceived. The algorithm involves computing empirical covariance of a potentially very large data set. Incremental update of the covariance matrix from time t' to time t

Table 3.1: Online Learning of Variable Stiffness

Input: $\mathbf{x}_t^d \forall t \in [0, T_f]$, S , \underline{k} , \bar{k} , $\underline{\sigma}$, $\bar{\sigma}$

- 1: **for** $t < T_f$ **do**
- 2: sense $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{g}(t)$
- 3: **if** teaching **then**
- 4: add deviation to data set: $\Xi \leftarrow \Xi \cup \{\tilde{\mathbf{x}}, t\}$
- 5: sort Ξ in order of increasing time
- 6: **end if**
- 7: update L_t and U_t (Eq. 3.1)
- 8: update moments based on previous values (Eq. 3.4)
- 9: compute Σ_t and its SVD $\mathbf{Q}, \mathbf{\Lambda}$
- 10: compute $\mathbf{\Gamma} \leftarrow \text{diag}([\gamma(\sigma_t^1), \gamma(\sigma_t^2), \gamma(\sigma_t^3)])$ (Eq. 3.3)
- 11: compute translational stiffness matrix $\mathbf{K}_t \leftarrow \mathbf{Q}\mathbf{\Gamma}\mathbf{Q}$
- 12: compute damping \mathbf{D}_t according to Eq. (2.26)
- 13: **end for**

follows directly from the additive form of the first and second moments:

$$\mathbf{M}_t^1 = \frac{N_{t'}}{N_t} \mathbf{M}_{t'}^1 + \frac{1}{N_t} \left(\sum_{i=U_{t'}}^{U_t} \tilde{\mathbf{x}}_i - \sum_{i=L_{t'}}^{L_t} \tilde{\mathbf{x}}_i \right) \quad (3.4a)$$

$$\mathbf{M}_t^2 = \frac{N_{t'}}{N_t} \mathbf{M}_{t'}^2 + \frac{1}{N_t} \left(\sum_{i=U_{t'}}^{U_t} \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^T - \sum_{i=L_{t'}}^{L_t} \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^T \right) \quad (3.4b)$$

and

$$\Sigma_t = \mathbf{M}_t^2 - \mathbf{M}_t^1 (\mathbf{M}_t^1)^T \quad (3.4c)$$

Pseudo-code for the learning procedure is given in Algorithm 3.1. In lines 3-6 the update of the data used for stiffness assignment is performed. It is vital that data points are only added to Ξ if teaching is performed, as the covariance would otherwise gradually decrease in the absence of perturbations, with the effect that the robot 'forgets' what it has been taught. In this work, we used no detection of teaching but let the teacher switch between two modes: teaching or not teaching. Lines 7-12 computes the the stiffness matrix based on the current window view of Ξ . A fixed rotational stiffness and critical damping (See Section 2.2.4) is used. A flow-chart illustrating how the perturbations are used during the control of the robot is shown in Fig. 3.2

3.2.2 EXPERIMENTS

Two experiments were conducted to evaluate the proposed system. The first is designed to demonstrate that the system can learn stiffness variations both in direction and magnitude, as instructed by the teacher. In the second experiment, we illustrate the usefulness of the system by teaching a stiffness profile for a task of pouring a drink into a glass.

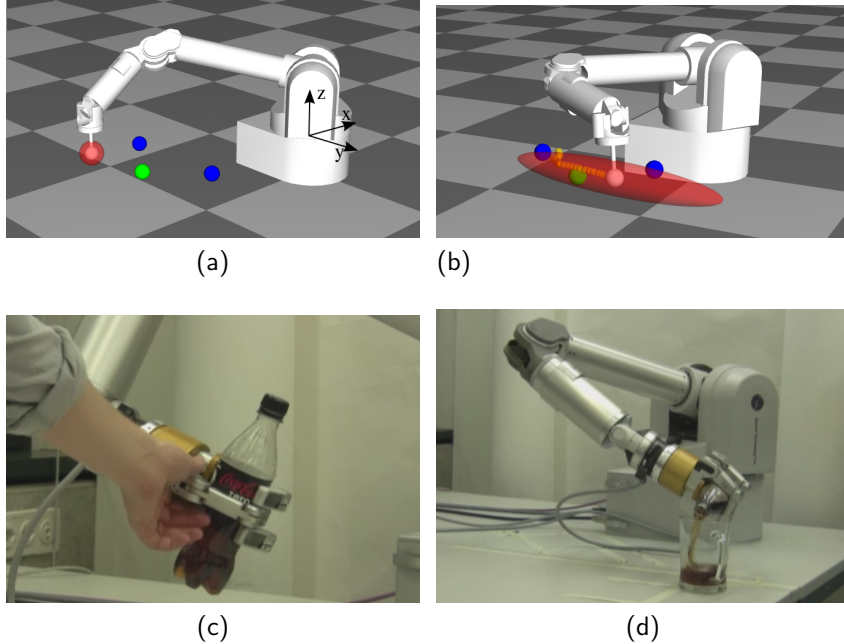


Figure 3.3: Top-left: The figure shows the layout of the via points for the first task. The points should be taken from left to right, with selective compliance in z-direction at the blue points and in x-direction at the green point, as indicated by the arrows. Figure top-right shows the graphical aid provided to the teacher while teaching. The simulator mirrors the robot motions while drawing a graphical representation of the current stiffness (the red ellipsoid). Figure bottom-left and bottom-right shows a snapshots of the robot pouring task during teaching and task reproduction.

SETUP

The system for learning stiffness through interaction as described in section 3.2.1 was implemented in a control module for a 7-dof Barrett WAM, using the RobotToolKit (RKT) and ROS software frameworks¹. The Cartesian impedance controller without inertia shaping described in Section 2.2.4 was used.

The rotational stiffness was set to a diagonal matrix with a constant rotational stiffness of 6 Nm/rad around all three axes. The lower and upper bounds for stiffness \underline{k} and \bar{k} were set to 50 and 350 N/m respectively. Empirically, 50 N/m is what the used setup needs to overcome static friction. The upper bound was set as a safety precaution. The parameters $\underline{\sigma}$ and $\bar{\sigma}$ were set to 0.005 and 0.05 respectively. These parameters control the values at which the the stiffness saturate and were set experimentally. The length S of the sliding temporal window was set to 1 second. The teacher could at the beginning of

¹RobotToolKit is an open source collection of tools for robot simulation and control developed by Eric Sauser. ROS (Robot Operating System) is open source robot middle-ware developed by Willow Garage

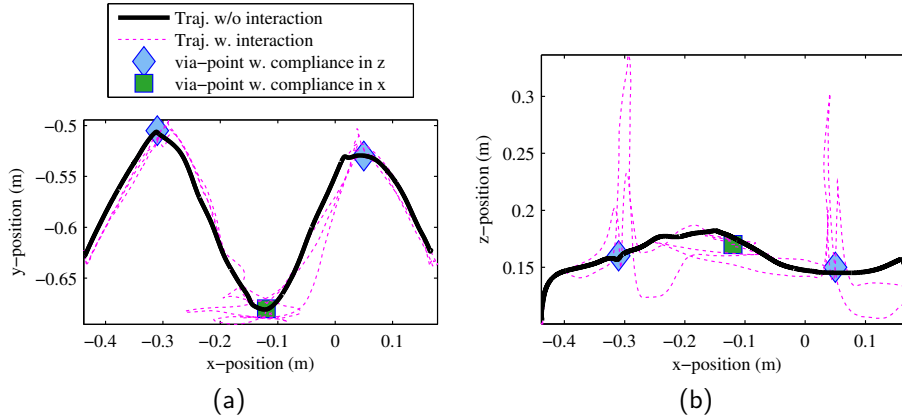


Figure 3.4: The figures show the reference trajectory with the trajectories resulting from the teacher’s interactions overlaid in the xy -plane and xz -plane respectively.

each task reproduction choose if teaching was to be performed or not (cf. line 3 in Alg. 3.1) by pressing a key on the keyboard.

As the focus of this work is not learning the kinematic task profile, simple record-and-replay was used for generating the desired pose trajectories. To this end, the robot was put in gravity compensation mode, and guided through the different motions by the teacher while the pose trajectory was recorded.

The translational stiffness was fed to the controller at a rate of 10 Hz. The reference point \mathbf{x}^d was updated at each iteration of the inner control loop, which runs at 500 Hz.

A RKT simulator was set up to provide graphical aid to the teacher while performing demonstrations by mirroring the robot movement on a screen and drawing a graphical representation of the current stiffness as an ellipsoid, see Fig. 3.3b. The ellipsoid is shaped inversely to the stiffness, so low stiffness in a direction is represented by the ellipsoid being elongated in that direction.

TASK 1: VIA-POINT TRAJECTORY

The purpose of this experiment is to illustrate the claim that the proposed system can learn stiffness variations in selective directions. The task consists in following a trajectory passing through three via points. Fig. 3.3 shows the via points in the robot’s work-space. The robot should be maximally compliant in the z -direction at the blue via-points, and maximally compliant in the x -direction in the other directions. This is a quantitative requirement in that the task constraints state specifically that the robot should assume its minimum allowed stiffness \underline{k} for one and one only of the singular value at each of the via-points. Furthermore, the directions of compliance at each via-point are specified to be aligned approximately² with the z -axis for the blue via-points and the x -

²This requirement is only approximate since it can not be expected by a human teacher to provide perturbations exactly aligned with a given direction.

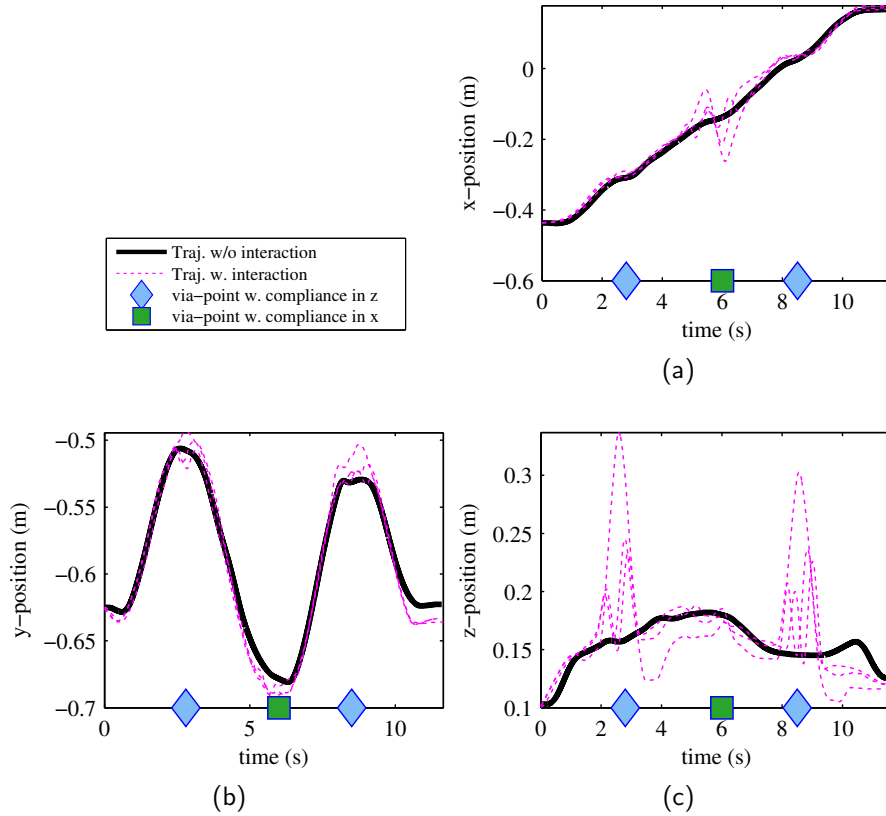


Figure 3.5: The figures show the reference trajectory with the trajectories resulting from the teachers interactions in each dimension over time. Note the variance near the via points.

axis for the green via-point. In between the via-points, the robot should stiffen up as quickly as possible³ to its maximal stiffness in all directions. The reference trajectory moves through the points with approximately constant speed. The total duration of the task is 12 seconds.

The robot executed the task three times while the teacher was providing input. Fig. 3.4a shows trajectory followed without interaction in the XY-plane of the base coordinate system, with the trajectories resulting from the teachers interaction overlaid. As is clear from the figure, the teacher imposed perturbations in the x-direction at the green via-point. Note also the small amount of variance imposed in y-direction at the blue via-points. These small perturbations are an unintended bi-effect of the larger perturbations imposed in z-direction at the same points, as can be seen in figure 3.4b. Fig. 3.5 shows the trajectories followed along each direction x,y,z of the base coordinate system over time. Even though the via-points were originally defined by space coordinates, they are implicitly anchored in time since the trajectories are time dependent. The times

³The rate at which the stiffness can change is limited as an effect of the sliding temporal window, refer to section 3.2.1

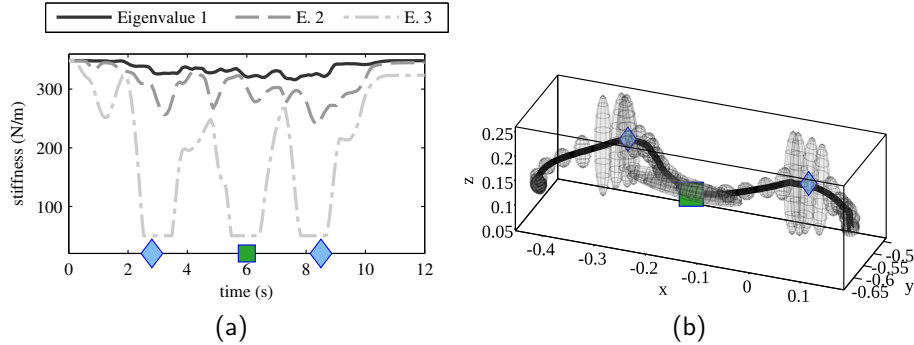


Figure 3.6: The figures show the stiffness trajectories resulting from the teachers interaction showed in Figures 3.4 and 3.5. **Left:** The singular values of the stiffness matrix over time. **Right:** The stiffness matrix visualized using compliance ellipsoids, with elongated direction corresponding to low stiffness.

at which the via-points are marked are simply the times at which the reference trajectory pass through these points. As can be seen in these figures, the trajectory from the teaching rounds generally has a bias error when compared to the reference trajectory. This bias is due to the teacher holding the end-effector and following along even when not imposing perturbations. Note that this bias does not affect the stiffness as only the covariance of the perturbations is used for determining the stiffness (cf. Alg. 3.1).

In Fig. 3.6a, the stiffness singular values resulting from the teaching is plotted over time. The plot clearly shows that maximum compliance is only reached in one direction at each of the via points. In Fig. 3.6a, the stiffness matrix is plotted as an ellipsoid at a series of points along the motion trajectory. The ellipsoids are built around the principal directions of the stiffness matrix, with low stiffness along a direction illustrated by the ellipsoid being elongated in that direction. As can be seen in this figure, the direction corresponding to the lowest singular value is approximately z-direction at the blue via-points and approximately x-direction at the green via-point. The human teacher being unable to impose perturbations exactly along the desired directions is the reason for the directions being only approximately correct.

TASK 2: POURING A DRINK

This task was chosen to show how the proposed system can be used to teach a realistic task that benefits from a varying stiffness profile. The task consists first transporting a bottle full of soda toward a glass. Once above the glass, the robot was to pour the drink. We state the following desired *qualitative* characteristics for this task:

1. During the reaching phase, the robot should be compliant in all directions, as position errors are not crucial and correcting for position errors with high stiffness can result in high accelerations of the end-effector which

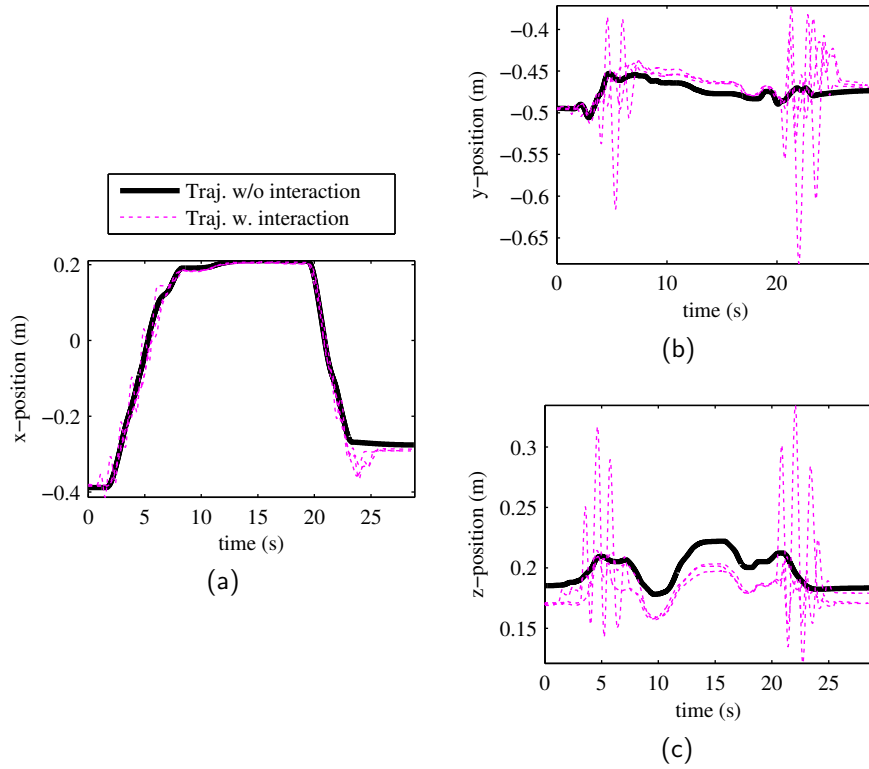


Figure 3.7: Figures (a)-(c) shows the trajectory followed by the robot when unperturbed along with the trajectory followed during two teaching rounds overlaid. Note that no perturbations were provided along the x-direction.

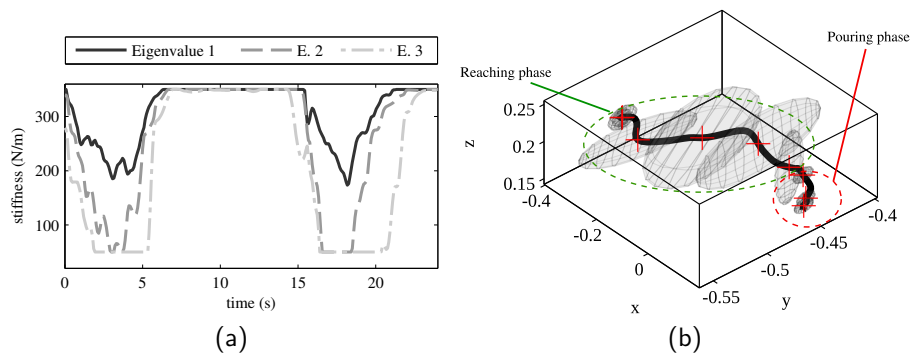


Figure 3.8: Figures (d) and (e) show the stiffness resulting from these teaching rounds. Note that all singular values reach their maximum during the pouring phase which takes place at second 10 to second 16. As is seen in figure (e), the principal directions in which the robot has reduced stiffness in the reaching phase lies in the yz-plane, while it remains fairly stiff along the x-axis.

spills the drink out of the bottle.

2. In the pouring phase, the robot should stiffen up in all directions, since the drink should be poured into the glass, even if moderately strong perturbations are encountered.
3. In the third stage, when the robot is reaching away from the glass, it is again desirable that a low stiffness is used, for the same reason as mentioned for the reaching stage.

The reference trajectory was acquired using record and replay. The total duration of the task is 25 seconds, and the critical pouring phase starts 10 seconds into the task and ends 6 seconds later.

The refinement of the pouring task consisted in decreasing stiffness in all directions in the reaching phase, letting the robot be stiff while pouring, and again decreasing the stiffness after the pouring phase. Since the stiffness was to be decreased in several directions at each point along the reaching phases, three rounds of teaching were performed. During each round, the teacher concentrated on introducing variance in along the three coordinate-axes x, y and z of the base coordinate system. The x, y, z components of the trajectories from the teaching rounds are shown in figures 3.7a, 3.7b and 3.7c. Note that the teacher did not impose as high variations in x -direction as in y , and z -direction. The reason for this is that the motion of the task was approximately aligned with the x -axis throughout the entire reaching phase. Perturbing the robot heavily along the planned direction of motion makes it hard for the teacher to respect the intrinsic time dependency of the task, thus risking to anchor the teaching in a part of the motion where this was not intended. Respecting the time dependency along other directions, especially those that are orthogonal to the direction of motion, is easier as the teacher can feel the robots desired motion and follow it while perturbing in other directions.

The result of the relatively smaller perturbations in the x -directions are directly visible in Fig. 3.8a, which shows the stiffness singular values resulting from the three teaching rounds. Clearly, all three values drop during the reaching phases, while two of them drop much more than the first. Fig. 3.8b shows the ellipsoid representation of the stiffness matrices for a subset of the points along the followed trajectory, again making it clear that the stiffness was principally reduced in the XZ -plane. As expected, the robot assumed a high stiffness at the beginning of the task and in the pouring phase, since the teacher provided no interaction there.

3.2.3 DISCUSSION

An online, incremental algorithm for learning variable stiffness has been introduced. The algorithm sets the stiffness negatively proportional to the covariance of perturbation data imposed by the teacher. The data taken into account for

determining the stiffness is taken from a sliding temporal window over the set of all provided data.

It has been assumed that the robot already knows a kinematic profile for the task. No assumptions have been made as to how this kinematic profile is generated. This means that the system can be used as an add-on to any other system learning robot motions. The single requirement is time-dependency, as this is a feature of the system presented in this work. Thus, both the position command and the stiffness command are explicitly anchored in time. This is limiting for tasks in which major perturbations from an unknown environment can be expected, e.g. manipulation tasks that involve significant contact forces.

In Section 3.2.1 it was explained that the presented system is data driven. The empirical covariance matrix of subset of the collected data is computed at each iteration. For our experimental setup, the computation required was well within the requirements of the 10Hz update frequency used for the stiffness. The real drawback of the data-driven approach is rather that all data points have to be saved in memory.

3.3 Learning Compliant Manipulation through Kinesthetic and Tactile HRI

This section reports on significant extensions and improvements of the interface presented in Section 3.2. The contents here are adapted from (Kronander and Billard, 2013). The first improvement is the introduction of a frequency domain separation of the measured interaction, where only interaction in a selected band have an effect on the stiffness. This has the benefits of reducing the effect of interaction components which arise from contact with the environment and that it allows the teacher to use slow interaction to feel the stiffness of the robot without actively changing it. In this work, we also generalize the interface so that it can be used for stiffness modulation both in Cartesian and joint space. We further introduce a new mode of interaction for *increasing* the stiffness. To this end, the robot monitors the pressure with which the teacher grasps the robot arm, and uses high grasping force as an indication to increase the stiffness, see Fig. 3.1. In Section 3.2, the position command came from prerecorded reference trajectories. Both the desired trajectory and stiffness variations were time-dependent. In this work, we show that our techniques for teaching stiffness variations can be used together with autonomous dynamical systems for modeling the positional aspects of the task, while encoding the stiffness variations as a function of the position of the robot rather than time. Which of the time-dependent or time-independent encodings to use depends on the task. Specifically, for tasks that require stiffness variations to occur in certain physical configurations, a time-independent encoding is advantageous since it is robust to perturbations delaying the execution of the task.

We demonstrate the proposed approach for learning compliant manipulation in two tasks. In the first experiment we revisit the pouring task from the previous section, but with a time-independent task representation. The proposed system for decreasing and increasing stiffness in Cartesian space is demonstrated. The second experiment demonstrates the system for teaching joint space stiffness variations in a match-lighting task. The results show that by teaching a varying stiffness profile, the success-rate when lighting a match is greatly increased compared to both fixed low and fixed high stiffness. Finally, the efficiency of the proposed teaching interfaces is evaluated in a user study, where subjects were asked to teach the robot a stiffness for improving the performance of a drawing task.

The addition of the filtering step, the option of choosing the interaction signal and the new modality for increasing the stiffness are illustrated Fig. 3.9, which highlights the differences between the interface presented here and OLVS in Section 3.2 as depicted in Fig. 3.2.

3.3.1 FILTERING THE INTERACTION SIGNAL

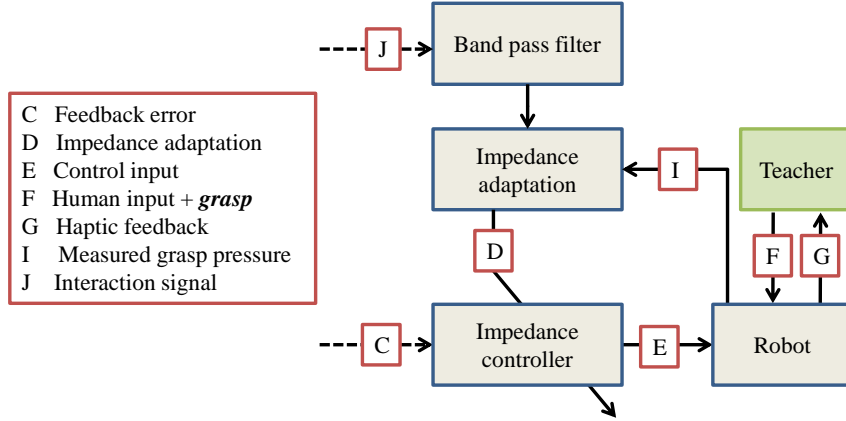


Figure 3.9: Flowchart highlighting the difference with the simpler version depicted in Fig. 3.2. The robot is controlled with either Cartesian or joint impedance control. The stiffness used by the controller is adapted online according to the interaction provided by the teacher. The interaction signal may be either position feedback error or measured interaction force/torque.

In Section 3.2, the position feedback error was used to monitor the interaction between the teacher and the robot. Alternatively, for robots equipped with joint torque sensors, e.g. the KUKA LWR, the sensed interaction torques can be used as interaction signal. This is often preferable to using the positional feedback signal as it allows interaction to be detected even if it does not result in movement of the robot away from its equilibrium position. This is important for applications in which the maximum stiffness is so high that humans will find it difficult to move the robot off its path. Here, we hence consider a more general abstraction called *interaction signal*. How this is measured is implementation dependent, and it will be clarified for each experiment exactly how the interaction is measured.

In OLVS (Section 3.2.1), perturbations in a time-window of a predetermined length are used to update the stiffness. Here, we incorporate a preparatory band-pass filtering step before proceeding to computing the stiffness using the filtered signal. The role of this filter is two-fold. First, it serves to remove high-frequency content from the interaction. This is desirable, as the raw signal is typically corrupted by high frequency sensory noise, and because interaction other than teaching (e.g. contact with environment) may contribute to the interaction signal. While separating the interaction signal in the frequency domain does not guarantee that such effects are avoided, it does make them less probable. Secondly, the lower frequency bound on the filter gives the teacher the possibility to use slow (low frequency) perturbations to feel how stiff the robot is without actively changing the stiffness. Choosing the lower cutoff frequency to $\underline{f} \approx 0.5$ Hz allows the teacher to comfortably perform slow perturbations to feel

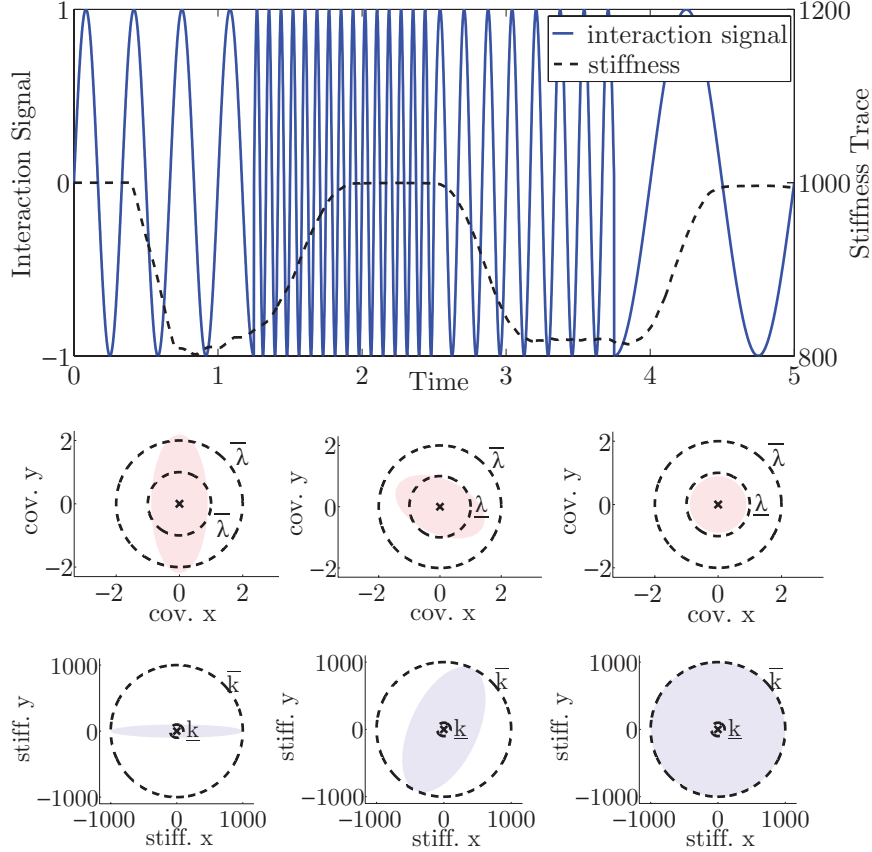


Figure 3.10: Top: An example of interaction signal and resulting stiffness trajectory. This is a sinusoidal interaction with a frequency that starts at 3 Hz, climbs to 12 Hz, drops to 6Hz and finally drops to 1Hz. Note how the interaction signal does not affect the stiffness for the frequencies outside the band-pass, chosen to [2Hz-10Hz] in this example. Middle: 2d examples of covariance of the interaction, Bottom: Stiffness ellipsoids resulting from the interaction examples.

the stiffness of the robot. The upper cutoff frequency \bar{f} should be set to a value slightly above the frequency that we can expect that a human teacher can be responsible for. There have been physiological studies on this topic, with reports of peak frequency for voluntary manipulation involving hand and arm movements in the range 4-8 Hz (Kunesch et al., 1989) and 2-5 Hz (Cathers et al., 1996) (Jones, 2000). To have some margin, the upper cut-off frequency was set to $\bar{f} = 10$ Hz in all experiments reported herein.

After the filtering step, the covariance for of the interaction signal is computed as in Eq. (3.2), considering $\tilde{\mathbf{x}}$ as the general interaction signal and not specifically the feedback error. Fig. 3.10 illustrates the effect of the filter as well as the relation between the covariance of the interaction signal and the stiffness.

3.3.2 STIFFNESS INCREASE BASED ON GRASP PRESSURE

With the perturbation-based interface, the teacher has the possibility to locally

reshape the stiffness matrix as well as decreasing its magnitude along one or more directions. High stiffness could only be achieved in the absence of interaction, which means that the default stiffness has to be equal to upper stiffness threshold. This is problematic, especially if the feedback position error is used as interaction signal, since a high default stiffness will require a significant physical work by the teacher for the robot to depart from its reference. To improve this aspect, we add a new interaction modality that allows to increase the stiffness. The teacher indicates increase in stiffness by increasing the grasp pressure with which he holds the robot. This way, the default stiffness can be chosen to a moderately compliant level, and increased or decreased locally as required.

Depending on the sensing available for detecting the grasp pressure, one can envision different ways to map the pressure to a selective increase in stiffness. For example, on a manipulator where several links are covered with artificial skin, the detected pressure on a link could be mapped to a stiffness increase selectively to its parent joint. Below, we describe the grasp pressure used in our CS implementation. For CS, we propose to map the perceived pressure from any part of the body to a uniform increase of all the singular values of the stiffness matrix. This allows to use the two interactive modes in conjunction to vary the stiffness at the end-effector, using perturbations to shape and decrease the stiffness, while using pressure to uniformly increase stiffness. To avoid interpreting skin pressure signals arising from collisions or other disturbances, approximately equal pressure on antagonistic parts of the arm is required for the stiffness to be changed. Let Ψ_t^a and Ψ_t^b denote the vectors of pressure detected on the agonist and antagonist parts of one part of the arm. Furthermore, let ψ_t denote a scalar which represents the grasp pressure of the arm at time t , defined by:

$$\psi_t = h(\Psi_t^a, \Psi_t^b) \frac{\max\{\Psi_t^a\} + \max\{\Psi_t^b\}}{2} \quad (3.5a)$$

$$h(\Psi_t^a, \Psi_t^b) = \begin{cases} 1 & \text{if } |\max\{\Psi_t^a\} - \max\{\Psi_t^b\}| < \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (3.5b)$$

The stiffness matrix is then shaped according to Eq. (3.3a), with singular values:

$$k_t^i = k_d + \frac{\bar{k} - k_d}{\bar{\psi} - \underline{\psi}} \max(\psi_t - \underline{\psi}, 0) - \frac{k_d - \underline{k}}{\bar{\lambda} - \underline{\lambda}} \max(\lambda_t^i - \underline{\lambda}, 0) \quad (3.6)$$

The first term is the default stiffness, which can be set to some value in the lower allowed stiffness range. The second term is the linear increase of stiffness from the default value k_d to the maximum allowed value \bar{k} when the pressure changes from its lower threshold $\underline{\psi}$ to its upper threshold $\bar{\psi}$. The lower threshold is important since without it, a normal grasp force used e.g. for wiggling the robot could unintentionally make the robot stiffen up. The upper threshold should be a value which is close to the maximum pressure a human teacher can

be expected to exert on the robot. Both these thresholds are easily tunable parameters. The third term is the decrease from k_d to \underline{k} as λ^i goes from its minimum threshold $\underline{\lambda}$ to its upper threshold $\bar{\lambda}$.

3.3.3 TASK LEARNING

The interfaces presented in Section 3.3.1 allow the teacher to modify the stiffness of the robot online. To learn from this interaction for subsequent autonomous task executions, a learning algorithm must be used. As for learning positional profiles, there are a wide range of algorithms in machine learning that can be used to learn a model based on demonstrations. In Section 3.2, we presented the Online Learning of Varying Stiffness (OLVS) algorithm for incrementally learning time-dependent stiffness variations using the perturbation-based interface for decreasing the stiffness. That approach did not learn the stiffness variations directly, but instead saved all the perturbations imposed by the teacher during the demonstration phase. These perturbations were then replayed during subsequent task executions, to achieve incremental refinement of the stiffness profile. In this work, we revisit the task of pouring a drink from Section 3.2.2, but instead learn the stiffness variations as a function of the position of the robot.

TASK: POURING A DRINK

This learning scenario illustrates how the interfaces described in Section 3.3.1 can be used in real world task. The task consists in first transporting a bottle full of soda toward a glass, and pour the soda into the glass when it has been reached. In this task, it is preferable to be very compliant in the reaching phase, where position is not important and stiff behavior may cause aggressive response to perturbations which result in spilling. However, when pouring, it is preferable to be stiff, since the robot should attempt to reject any external perturbations which may move the bottle from the glass. The robot used for this task is the 7 DoF Barrett WAM. The forearm of the robot is covered with artificial skin of the type presented in (Cannata et al., 2008), see Figures 3.1 and 3.13.

Our previous treatment of this task in Section 3.2.2 used fixed reference trajectory for the position and orientation, and OLVS for learning the stiffness variations. The reference trajectories as well as the learned stiffness profile were thus time dependent, which meant that perturbations that significantly delayed the task would result in failure such as the drink being poured before the glass was reached. Below, we revisit the pouring task and address these shortcomings by using autonomous dynamical systems for modeling the kinematics of the task. Furthermore, the stiffness is learned as a function of the robot position, which ensures that the stiffness variations happen when the task requires it rather than at a specific time.

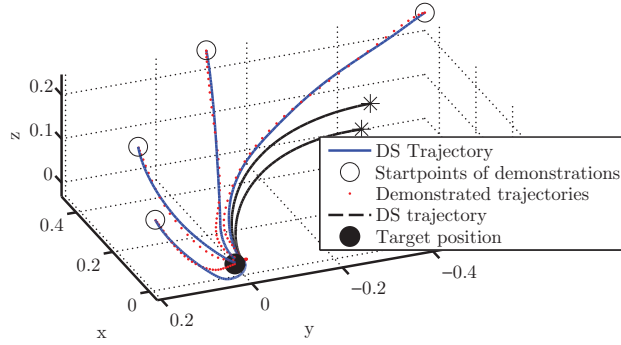


Figure 3.11: The figure shows the demonstrations provided by the teacher (red dotted line) and the DS trajectories generated from the same starting positions (blue). Two trajectories from other starting positions (black) illustrate that similar motion is reproduced, with convergence to the target.

Position profile

To learn a generalized model of the kinematics of the task, we model the position profile as an autonomous Dynamical System (DS). The DS is modeled by a Gaussian Mixture Model, trained with the Stable Estimator of Dynamical Systems (SEDS) (Khansari-Zadeh and Billard, 2011) algorithm to guarantee that the motion converges to the glass regardless of starting position. This approach uses a set of demonstrations to optimize the parameters of a GMM (refer to Section 2.2.5) encoding an estimate of the joint probability of the position and the velocity of the robot. The velocity then becomes a function of position via GMR on the learned GMM. The representation of the desired position profile as a DS is illustrated in Fig. 3.11.

Note that during task execution, we do not use continuous feedback of the actual position to the DS as this would yield a system completely without any apparent stiffness. Instead, we define an internal state variable $\mathbf{y} \in \mathbb{R}^3$ which will be updated using the DS and which is related to the reference trajectory as follows. At the onset of motion, the position of the end-effector, $\mathbf{x}_0 \in \mathbb{R}^3$ with respect to the target position $\mathbf{x}_0^* \in \mathbb{R}^3$ is computed:

$$\mathbf{y}_0 = \mathbf{x}'_0 - \mathbf{x}_0^* \quad (3.7a)$$

The desired velocity w.r.t the target is then computed using the DS, which maps the current relative position to the desired relative velocity through the non-linear function $\mathbf{f} : \mathbb{R}^3 \mapsto \mathbb{R}^3$:

$$\dot{\mathbf{y}}_t = \mathbf{f}(\mathbf{y}_t) \quad (3.7b)$$

To find a set point \mathbf{x}_{t+1}^r for the controller, we integrate the previous relative position and change coordinates to the robot reference frame via addition of the

current target position:

$$\mathbf{x}_{t+1}^r = \mathbf{y}_t + \Delta t \dot{\mathbf{y}}_t + \mathbf{x}_t^* \quad (3.7c)$$

Above, Δt denotes the length of one iteration of the control loop ($\delta t = 0.002$ in our implementation). The target position is updated from its actual position⁴, while the robot’s position is integrated open loop:

$$\mathbf{y}_{t+1} = \mathbf{x}_{t+1}^r - \mathbf{x}_{t+1}^* \quad (3.7d)$$

This allows to use the robustness of the DS w.r.t to a moving target, but perturbations on the robot state are completely handled by the impedance controller.

Orientation Profile

When pouring a drink, the bottle should ideally remain vertically oriented everywhere except when it is correctly positioned above the glass. Having a time dependent orientation profile will cause the robot to pour the drink outside the glass if the robot is unable to reach the glass in the planned time. The control of the orientation should therefore depend on the state of the reaching motion, and exhibit coupling so that if the end-effector is moved away from the glass, the orientation of the bottle immediately gets corrected. In (Shukla and Billard, 2011), such coupling between reach- and grasp motions was observed in human subjects. In this work, we use a simple such scheme, which couples the orientation of the end-effector to the state of the reaching motion by using distance between the end-effector and the glass as coupling variable.

We adopt the unit quaternion representation for the orientation of the end-effector. Let \mathbf{r}^a and \mathbf{r}^b denote the quaternions corresponding to the default and pouring orientation respectively. These orientations were captured by moving the robot to the desired orientations and recording their quaternions. To compute the desired orientation at time t , \mathbf{r}_t^d , we interpolate between \mathbf{r}^a and \mathbf{r}^b using spherical linear interpolation:

$$\mathbf{r}_t^d = \frac{\sin(\Omega(1-w))\mathbf{r}^a + \sin(\Omega w)\mathbf{r}^b}{\sin(\Omega)} \quad (3.8a)$$

where Ω satisfies $\cos(\Omega) = \mathbf{r}^a \cdot \mathbf{r}^b$, and $w \in [0, 1]$ is the interpolation parameter, which we define as function of the difference of the current position of the end-effector \mathbf{x}_t and the position of the glass \mathbf{x}_t^* :

$$w(\mathbf{x}_t^r, \mathbf{x}_t^*) = \exp(-\kappa(\mathbf{x}_t^r - \mathbf{x}_t^*)^T(\mathbf{x}_t^r - \mathbf{x}_t^*)) \quad (3.8b)$$

Here, κ is a parameter that controls how quickly the rotation is changed as a function of movements to and from the glass. This parameter was set experimentally to $\kappa = 0.06$.

⁴The position of the glass is tracked at 200 Hz by a marker-based vision system.

Stiffness Profile

The desired stiffness profile in the pouring task is to be compliant during transport and stiff when pouring. As in the choice of end-effector orientation, the stiffness should thus depend on the state of the reaching motion rather than time. This is a common situation for manipulation tasks, i.e. that there is some kind of free motion where it is generally preferable to use a low stiffness, and a manipulation step that may involve contact that requires a specific stiffness to be used locally. Therefore, it is natural to learn the stiffness variations as a function of the relative position between the robot and the object of interest. Furthermore, the teaching process can be simplified by first demonstrating the general stiffness that the robot should use as its default, and in a secondary step demonstrating the local changes from this default stiffness as the robot is performing the task.

We used the combined perturbation and pressure based interface described by Eq. (3.6). First, the default stiffness k_d was set to the value in the middle of the allowed stiffness range $[\underline{k}, \bar{k}]$. In our implementation, this corresponds to $\bar{k} = 1000$ N/m, $\underline{k} = 70$ N/m and $k_d = 535$ N/m. For demonstrating the default stiffness, the robot was set to hold a constant position while the teacher was interacting with it to find a satisfactory default behavior. Since a uniformly low default stiffness was desirable, the teacher provided spherical perturbations with increasing amplitude until a sufficiently low stiffness was achieved. Once the interaction stopped, the resulting stiffness matrix was saved as default stiffness. The trace of the end-effector and the final stiffness at the end of the first teaching step are shown in Fig. 3.12.

The robot then executed the task with the combined perturbation and pressure-based interface (Section 3.3.2), but now with $k_{def}^i, i = 1, 2, 3$ according to the default stiffness learned in the preceding step. The orientation of the default stiffness matrix is described \mathbf{Q}_{def} . To preserve this orientation, the computation of the perturbation covariance matrix, Eq. (3.2), is preceded by rotation of the perturbation data points onto the basis given by the columns of \mathbf{Q}_{def} . During task execution, the teacher grasped the forearm of the robot and applied pressure as the robot approached the glass, see Fig. 3.13. Three such demonstrations were provided, and the stiffness trace was recorded when interaction was detected together with the relative position between the end-effector and the glass $\mathbf{x} = \mathbf{x}^r - \mathbf{x}^*$.

Based on the recorded data set, we seek a functional relationship that allows the robot to choose its stiffness based on the current relative position to the glass. To minimize the amount of parameters used for the modeling of this relationship while ensuring that the resulting stiffness is always symmetric and positive definite, we represent the stiffness matrix by its Cholesky vector $\mathbf{l} =$

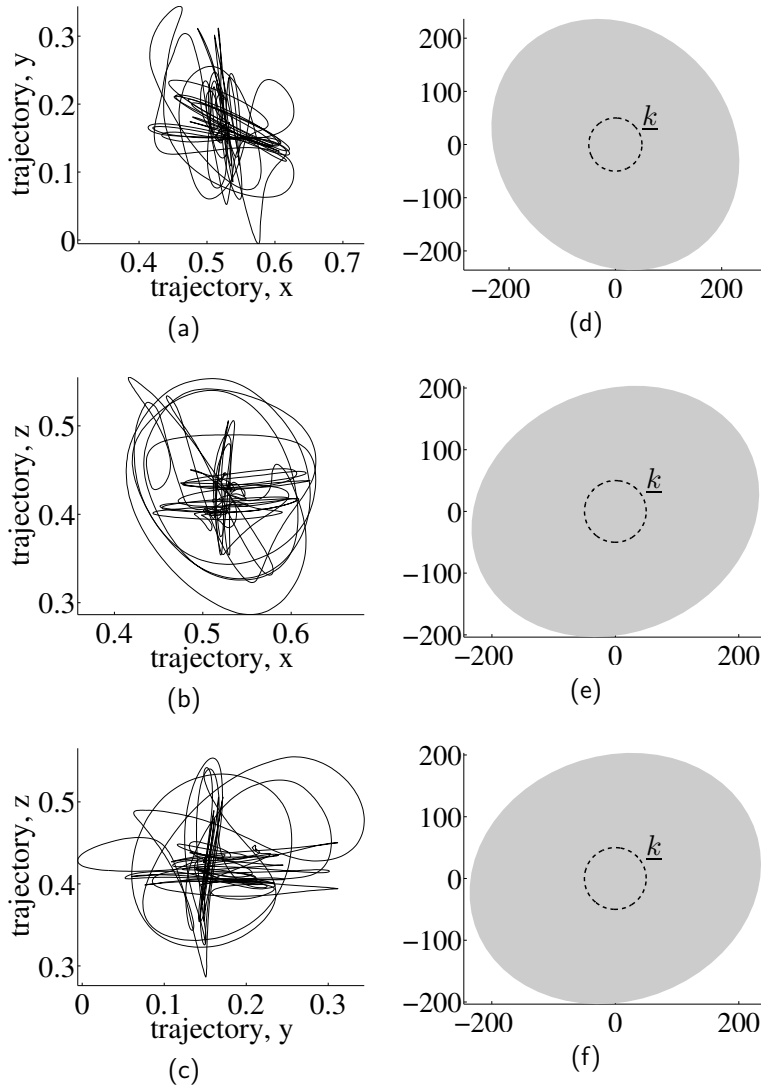


Figure 3.12: Top: the end-effector trajectory (left: XY, middle: XZ, right YZ) during interaction for determining the default stiffness. Bottom: resulting stiffness ellipses (left: XY, middle: XZ, right YZ)

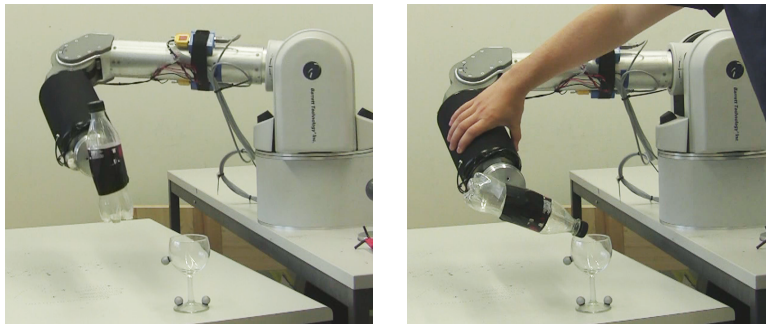


Figure 3.13: Teaching the stiffness variations for the pouring task. Here, the robot uses the default, low stiffness learned in the preceding step, and stiffens up locally when the teacher applies pressure on the skin.

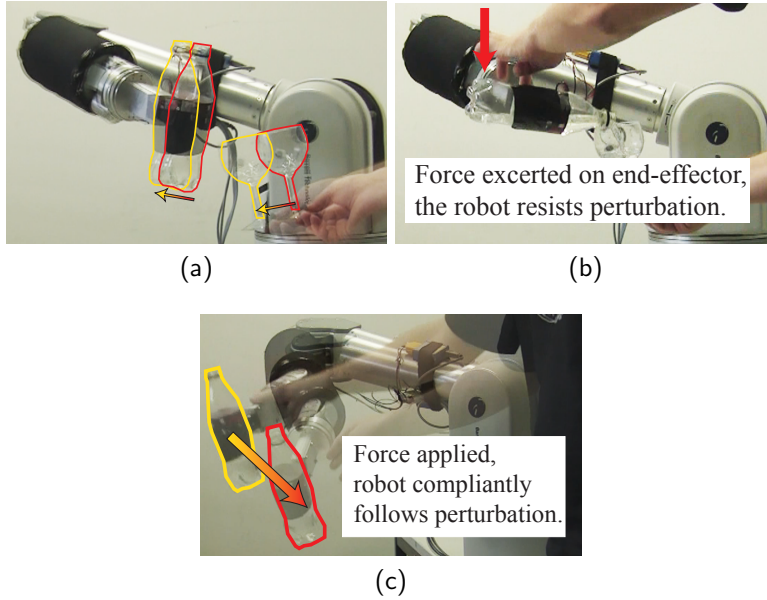


Figure 3.14: **a):** Illustration of online motion adaptation as the glass is moved. **b):** The robot resists the perturbation when pouring and stiffly remains in position. **c):** The robot is compliant when it is far from the glass.

$[L_1, \dots, L_6]^T$, defined by⁵:

$$\mathbf{K} = \mathbf{L}^T \mathbf{L}, \quad \mathbf{L} = \begin{bmatrix} L_1 & L_2 & L_3 \\ 0 & L_4 & L_5 \\ 0 & 0 & L_6 \end{bmatrix} \quad (3.9)$$

To represent the demonstrated data in a compact model, we chose to use a GMM. Since the complexity of the data is expected to be low, we set the number of Gaussians to two only. The GMM was then trained using the standard Expectation Maximization (EM) algorithm (Dempster et al., 1977). The resulting GMM is a joint density estimate of the vector $\boldsymbol{\xi} \in \mathbb{R}^9$ concatenating the relative position between the end-effector and the glass and the Cholesky vector:

$$p(\boldsymbol{\xi}) = \sum_{k=1}^K \pi^k \mathcal{N}(\boldsymbol{\xi}; \mathbf{m}^k, \mathbf{C}^k) \quad (3.10)$$

During task execution the robot should use this model to choose its stiffness Cholesky vector as a function of the relative position to the glass. This can be done by GMR, which computes the conditional expectation $E\{\mathbf{l}|\mathbf{x}\}$ based on the GMM. One of the properties of standard GMR is generalization, which means that relationships found locally in the training data can be used when inferring distant points in the input space. For many applications, this is a desirable effect since it can reduce the quantity of training data. However, in some situations it

⁵Note that \mathbf{K} here represents the translational part of the stiffness matrix.

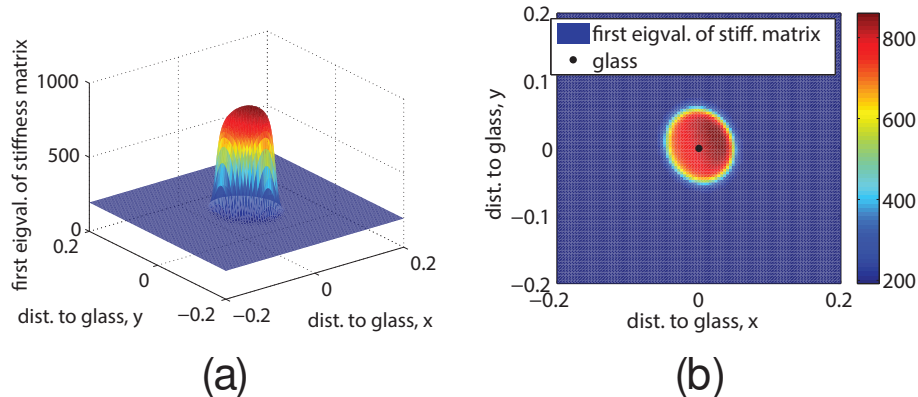


Figure 3.15: Left: The plot shows a 3d plot of the maximum singular value of the stiffness matrix as a function of the distance to the glass in the xy-plane. Right: Same data as in the left plot, with a color-map for visualizing variations of the maximum singular value of the stiffness matrix.

is inappropriate to generalize in this manner. The case of inferring stiffness is an example of such a situation. Generalizing a locally linear relationship to regions outside the training range would mean that the inferred stiffness could grow unbounded. To prevent this, we propose a novel regression strategy in which the generalization can be controlled with a single parameter. This regression method is described in detail in Appendix A.1. In addition to the GMM, this method makes use of a basic conditional distribution $p_b(\mathbf{l}|\mathbf{x})$ that describes the desired relationship between inputs and outputs in regions poorly covered by the training data. As explained in Appendix A.1, for regression it is sufficient to define the mean of this basic distribution. In this particular application, the mean of the basic distribution is the Cholesky vector of the default stiffness from the first demonstration phase.

As the robot performs the task autonomously, the Cholesky vector $\mathbf{l} = \mathbf{l}(\mathbf{x})$ is found with the regression method described in Appendix A.1. The output from the regression is then used to reconstruct the Cholesky matrix which is finally used to compute the stiffness according to Eq. (3.9). Fig. 3.15 shows how the maximum singular value of the stiffness matrix varies as a function of the distance to the glass in the xy-plane. Clearly, the desired behavior of the stiffness increasing only locally when the robot is close to the glass is fulfilled. We plotted only one of the singular value, since the pattern is similar for all three⁶.

TASK: LIGHTING A MATCH

Lighting matches is a typical example of a task that humans perform with

⁶In this task, a uniform increase was desired, and the pressure-based interface increases the stiffness uniformly.

ease, whereas a position controlled robot will either break the match or apply too little force if the matchbox is not exactly in the programmed position. One solution would be to control the contact force between the match and the matchbox directly. This would, however, require very accurate force sensing at the wrist and a more complicated control system, with more parameters to be tuned. We hypothesized that decreasing the stiffness as the robot strikes the match is sufficient to limit the contact force in this task.

Position profile

A JS control architecture was chosen for this task. The primary reason for this choice was to exemplify how our system can be used in JS⁷. One demonstration of a joint space trajectory was given using kinesthetic teaching. To get a smoothed version of the demonstrated trajectory that is able to reproduce the trajectory with high precision, we employed a Gaussian Process (GP) to encode the demonstrated motion. For reproducing the motion, the reference position \hat{q}_t^i of each joint i at time t is retrieved through Gaussian Process Regression (GPR), see Section 2.2.5.

Stiffness profile

In order to showcase the need for varying stiffness, we first carried out a series of trials with fixed stiffness values. In the first series of trials, the joint stiffness was set to a high, constant value of 1000 Nm/rad for all joints. This stiffness is similar to position control, as the robot is very stiff and effectively rejects most external forces. 20 trials were carried out, and most common outcome of these attempts was that contact was established but the match broke. In a second set of trials, the stiffness of the elbow joint was set to a constant value of 50 Nm/rad. Motion around the elbow corresponds in this case to approximately perpendicular motion between the match and the matchbox, hence reducing this stiffness reduces the contact force between the match and the matchbox⁸. The lower the stiffness in this joint is set, the lower the contact force, and 50 Nm/rad is the minimum value that was considered acceptable from the perspective of maintaining the possibility to track the position trajectory. Again, 20 trials were carried out, with the most common outcome being that the match was struck through thin air, due to inadequate precision in positioning the match before the striking motion.

To achieve high precision in positioning while ensuring compliant motion when striking, the stiffness should be decreased only locally in the striking phase. A JS stiffness profile was taught using the output from the joint torque sensors of the KUKA LWR as interaction signal. This experiment did not make use of the pressure-based interaction modality for increasing the stiffness, as no skin

⁷This task could also have been encoded in CS.

⁸The contact force arises from the controller trying to reject the perturbation of the planned trajectory of the match as it comes in contact with the matchbox. The lower the stiffness, the lower the controller effort and hence the contact force.

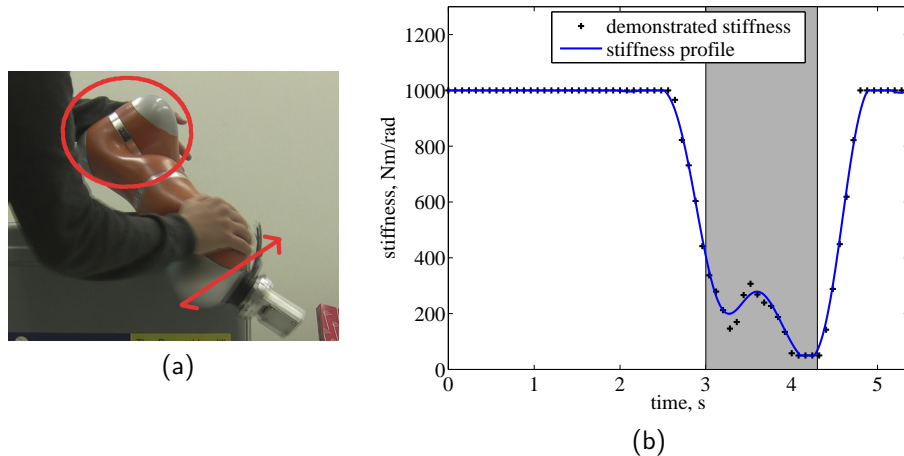


Figure 3.16: Left: The demonstration of stiffness variations. The teacher indicates decrease of stiffness of the elbow joint (encircled) during the striking motion. The red arrow indicates the direction of wiggling. Note that the demonstration was carried out without a match mounted on the robot. **Right:** The stiffness profile for the elbow joint in the match-lighting task. The shaded area indicates the striking phase. The remaining 6 joints of the robot had a constant stiffness of 1000 Nm/rad.

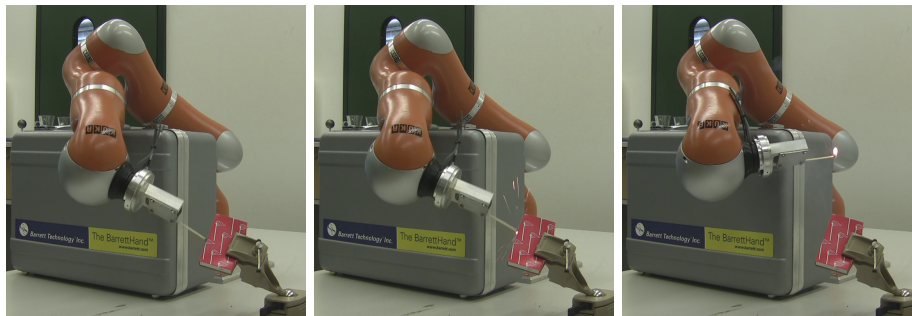


Figure 3.17: Snapshots from task execution where the robot successfully lights the match using the learned stiffness profile.

modules were available for the platform. While the robot executed the motion, the teacher intervened in the striking phase by wiggling the elbow joint around its reference trajectory, see Fig. 3.16a. The resulting stiffness was recorded with corresponding time stamps. For smooth reproduction of the stiffness profile, we used a GP similarly to the joint position trajectory. The resulting stiffness profile can be seen in Fig. 3.16b.

The result was accurate positioning before establishing contact with the matchbox, and a compliant motion through the striking phase. While this controller still occasionally breaks the matches, the success rate is significantly higher (refer to Table 3.2) than for the two settings with fixed stiffness values. Large variability in the quality of the matches as well as small positioning errors of the matchbox are believed to be the most important reasons of failure.

Table 3.2: Rates of success the match lighting task with constant high stiffness, constant low stiffness in the elbow joint and learned varying stiffness in the elbow joint. A total of 20 trials was carried out for each case.

	Broke	Broke and was lit	Not lit	Lit	Rate of success
const. high stiff.	4	11	2	3	15%
const. low stiff.	1	3	14	3	15%
learned var. stiff.	0	2	1	17	85%

3.3.4 USER STUDY

In the previous Section, we described and exemplified how our interfaces can be used for learning the stiffness for different tasks. A crucial aspect of any LfD method is how convenient it is for non-roboticists to use it. Indeed, the whole motivation of LfD is to allow people without technical knowledge in robotics to transfer skills to the robot. In this section, we address this by evaluating our approach through a user study. To provide a simple alternative to our system, we developed a Graphical User Interface (GUI) with which the user can change the stiffness along the three Cartesian axes using sliders.

SUBJECTS

Two groups of 14 participants took part in the study. The participants in one group were asked to select the stiffness using the GUI. The participants in the second group used our proposed approach, combining wiggling motions for selectively decreasing the stiffness with grasp force for uniformly increasing it as described in Sections 3.3.1 and 3.3.2. In the following, the two groups will be referred to as the GUI group and the PHRI (Physical Human-Robot Interface) group. The subjects were mainly recruited on campus at EPFL. They were naive to robotics with an age-range of 19-35. 10 out of the 28 participants were female, with equal gender distribution in each group.

TASK

The task considered was chosen specifically such that there are different stiffness requirements along different directions in task space. The task consisted in drawing a straight line on a white-board. While the robot was drawing, a series of impulses were added to the control output, acting equivalently to external perturbations. When such an impulse was applied, the robot departed from its straight line trajectory. Furthermore, the reference trajectory was deliberately placed behind the contact surface, such that a contact force was established between the pen and the board, see Fig. 3.18. For safety reasons, the robot was programmed to abort the task whenever the contact force exceeded 15N. For maximum performance in this task, the stiffness along the x-axis (see Fig. 3.18) should be high, so that the robot can counteract the force impulses. To ensure a low contact force, the stiffness along the z-axis should be low. The stiffness along the y-axis has no influence on the task performance and could hence take arbitrary values. The stiffness was set to be constant throughout the task.

EXPERIMENT PROTOCOL

Each subject was given a description of the task, and the instruction to choose the stiffness of the robot so as to minimize the contact force and the departure from the straight line. The instructions did not reveal the correct task stiffness described above. Subjects were also given a description of the interface (GUI or

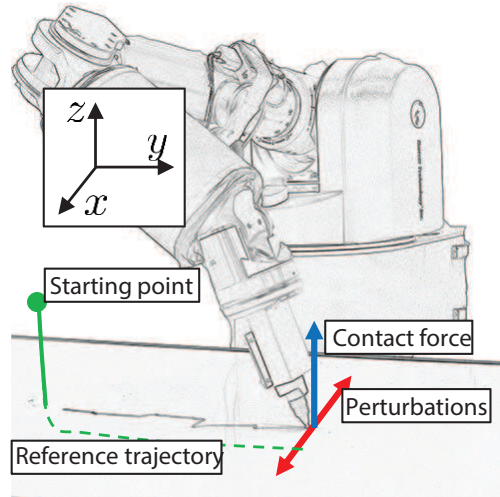


Figure 3.18: The figure illustrates the drawing task. Force impulses along the x-axis cause the robot to depart from the desired, straight trajectory. The dashed line represents the part of the reference trajectory that lies below the contact surface, giving rise to a contact force along the z-axis. The teacher's role is to change the stiffness of the robot, such that the perturbations are rejected while the contact force remains small.

PHRI depending on the group). After reading the description, the subjects were allowed to familiarize themselves with the interface during a practice period of two minutes. During this time, the robot maintained a static reference position while the participants were allowed to use the interface and interact with the robot to understand how their input affected the stiffness of the robot. The subjects were also allowed to ask the experiment supervisor interface-related questions during this time.

After the two minutes of practice, the subjects were asked to improve the performance of the drawing task by changing the stiffness of the robot. During the teaching process, they could at any time ask the experiment supervisor to start a new trial of the task. Subjects were allowed to change the stiffness while the robot was stationary between trials, or during the trials. They were also allowed to temporarily remove the pen if they wished to do so. Questions relating to the task (e.g. what effect could be expected by increasing or decreasing the stiffness) were not answered, while questions regarding safety (e.g. where to hold the robot) were.

A successful trajectory was defined as follows:

1. The contact force never exceeded 15N, i.e. the robot did not abort the drawing before reaching the end of the line.
2. The standard deviation of the trajectory along the x-axis was below a predefined threshold value.

Table 3.3: Objective and Subjective results from the user study. The table entries are formatted as mean (standard deviation) for each of the performance measures.

	Teaching time	Number of trials	SUS	TLX
PHRI	239.8 (100.1)	5.4 (2.2)	68.2 (16.7)	34.4 (16.9)
GUI	467.9 (260.7)	11.0 (6.9)	69.4 (15.6)	43.92 (17.3)

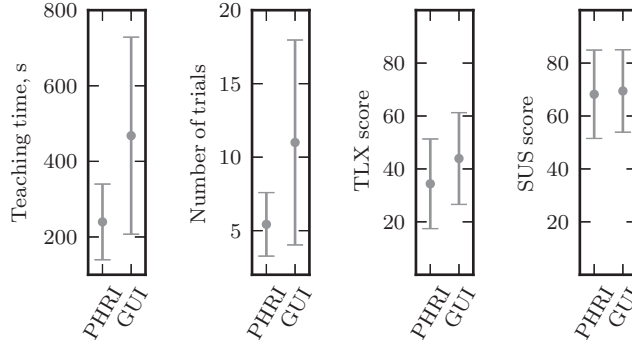


Figure 3.19: The figures show the mean and standard deviation of the performance measures for the two groups.

The subjects were informed about their progress only once they had managed to tune the stiffness such that a successful trajectory had been completed.

After completing the teaching, the subjects were asked to fill out a digital questionnaire containing the NASA Task Load Index (TLX) (Hart and Staveland, 1988) and the System Usability Scale (SUS) (Brooke, 1996) forms. The NASA Task Load Index is a commonly used evaluation method for the workload of a task. The evaluation consists in a series of workloads (e.g. physical, mental and temporal demand) which are rated using a 20-point Likert-scale. The System Usability Index is based on 5-point Likert-scale agreement response to statements about the user-friendliness of a system, e.g. ‘I thought that the system was easy to use’. More details about SUS and TLX are given in appendices A.2 and A.3 respectively.

3.3.5 RESULTS

Our main hypothesis was that teaching stiffness variations using the PHRI would be more efficient than using the GUI, which does not provide the immediate haptic feedback that the PHRI does. We could observe that subjects in the GUI group needed haptic feedback to understand the effect of a change in stiffness value. Indeed, the vast majority of the subjects in the GUI group conducted their teaching by changing a value on the GUI and subsequently pushing or pulling the end-effector in different directions to get an idea of the effect that

the change in stiffness brought.

As quantitative measures of teaching efficiency, we used the total amount of trials and the total teaching time⁹ before the subjects had completed a successful trajectory. These measures are presented in Table 3.3 along with the subjective results from the TLX and SUS questionnaires. The mean and standard deviation for each measure is also plotted in Fig. 3.19. The PHRI is clearly advantageous in terms of the time it took the participants to teach the task (One-way ANOVA: $p=0.006$; Welsh: $p=0.004$; Mann-Whitney: $p=0.02$). It also required fewer trials (One-way ANOVA: $p=0.01$; Welsh: $p=0.007$; Mann-Whitney: $p=0.03$). Regarding usability, the GUI and the PHRI are on par with no statistically significant difference between the SUS-scores of the two groups (One-way ANOVA: $p=0.84$, Welsh: $p=0.42$, Mann-Whitney: 0.47). While the PHRI achieved a lower value of TLX than the GUI (the lower the TLX, the better; range 0-100), this difference is not enough to give a statistically significant advantage to the PHRI (One-way ANOVA: $p=0.16$; Welsh: $p=0.08$; Mann-Whitney: $p=0.08$). A pairwise comparison for each of the factors underlying the SUS and TLX scores was also carried out, but no statistically significant difference was found.

From the results of the user study, we draw three conclusions:

1. All subjects, while being naive to robotics, were able to very rapidly figure out how to use each interface and to determine the appropriate stiffness for the task.
2. Both interfaces were perceived as user-friendly (An interface with a SUS-score above 65 is generally considered user-friendly).
3. The PHRI was the most efficient of the two interfaces, as it allowed the users to teach the task faster and with fewer trials.

3.3.6 DISCUSSION

We have addressed the problem of teaching tasks that require or can benefit from task-based varying stiffness. We proposed two modes of interaction and presented interfaces that adapt the robot stiffness based on these interactions. To make the robot more compliant, the teacher wiggles the robot around its planned trajectory as it is performing the task. To make the robot more stiff, the teacher increases the pressure with which he holds the robot. The robot responds online by adapting the stiffness based on this interaction, providing the teacher with instant haptic feedback on the effect of the teaching. We have exemplified and demonstrated the usefulness of our approach in both CS and JS, and validated the efficiency of the proposed interface through a user study.

⁹The total teaching time included the trials and the time for adjusting the stiffness while stationary in between trials.

In the second task (Section 3.3.3), we exemplified the system for teaching joint-space stiffness variations in the challenging task of lighting a match with the KUKA LWR. We showed that the rate of success for lighting matches was significantly increased by varying the stiffness compared to both constant high and constant low stiffness. None of the task coordinates CS and JS are universally better than the other. Which one to use depends on the task and the robot at hand.

As opposed to other approaches to tuning the compliance parameters in manipulation, our approach does not use a model of the environment, nor does it require to optimize over a cost-function describing the task. Instead, it assumes that the teacher is capable of demonstrating appropriate stiffness variations to the robot. For slow manipulation tasks, it is easy for a human to determine when the robot is too stiff or too compliant. This was shown by the results of our user study, in which all subjects were able to figure out how the stiffness should be adapted to increase the task performance, regardless of the interface (GUI or PHRI) that was used for teaching. For tasks that involve high speed motion, it can be difficult for the teacher to have the time to achieve the desired stiffness variations along the trajectory. A straightforward remedy to this is to slow down or pause the motion during the teaching process. The use of incremental learning can also make it easier for the user to gradually transfer the appropriate stiffness profile. However, our interface (or any other instruction-based interface) is likely not applicable to highly dynamic tasks where varying stiffness is required to achieve explosive motions¹⁰, such as kicking a ball. For such tasks, the optimal stiffness control involves fast switching between saturation levels (Garabini et al., 2012), which is problematic to demonstrate with the proposed interfaces as the possible rate of change is limited. Furthermore, it is unlikely that even an expert would be able to determine near-optimal stiffness variations for such tasks.

Varying stiffness is often important in tasks involving tools in physical contact. For such tasks, the wiggling motions to decrease the stiffness may not be possible. One solution to this difficulty is to let the robot execute the task without contact during the teaching phase. This can be achieved in several ways, e.g. by adding an offset to the reference trajectory or simply by removing the tool during teaching (as we did in the match-lighting task in Section 3.3.3).

3.4 Conclusion

In this chapter, we have highlighted that many tasks require or can benefit from being executed with a varying stiffness profile. We proposed two interaction modalities that allows lay users teach a varying stiffness profile easily. The usability was confirmed through a user study.

¹⁰This type of motions can be achieved with passively compliant joints, which have the possibility to store energy.

We have focused on acquiring the stiffness profile for a task. However, varying stiffness profiles can potentially destabilize the closed loop control system. This issue has been largely ignored in related work as well as in this chapter. The next chapter addresses this stability issue.

STABILITY ISSUES IN VARYING IMPEDANCE CONTROL

4.1 Introduction

As detailed in Sections 2.1.3 and 2.1.1, there are several open issues related to the link between learning from demonstration and impedance control. It is clear that many tasks require varying stiffness, and many others can certainly benefit from it. However, a critical open issue in this regard is the stability analysis of a system controlled with a varying stiffness. This chapter provides a new, state-independent stability condition relating the stiffness and damping and their time derivatives. The stability condition applies widely in impedance control with *constant inertia and varying stiffness and damping* and is not limited to any particular means of defining that impedance profile. The work presented in this chapter is at the time of this writing under review for publication in a robotics journal.

Variable impedance control can be seen as a special case of gain scheduling, which is a common technique in control for applying linear control design techniques to nonlinear systems. In gain scheduling, it is generally difficult to theoretically guarantee stability, and loose statements regarding stability with sufficiently small or slow gain variations are commonplace in the literature (Shamma and Athans, 1992). The success that variable impedance control has already had in numerous applications is an indication that reasonable impedance variations are sufficiently small/slow for a wide range of tasks. Nonetheless, these notions are of limited use when designing varying impedance profiles since there is no way of knowing how slow or how small the variations need to be.

There is hence a need for analysis or control methods that can guarantee stable execution of variable impedance tasks. This issue has been recently addressed in (Ferraguti et al., 2013), wherein a tank-based approach to passive varying stiffness was proposed. Their system uses the total energy of the manipulator (kinetic plus virtual potential energy coming from stiffness term), but does not constrain this function to strictly decrease. Instead, any dissipated energy is added to a virtual energy tank, a concept originating from (Duindam and Stramigioli, 2004; Stramigioli et al., 2005), from which energy can be extracted in order to implement stiffness variations. The tank is given an initial level of energy and a maximum allowed level of energy. These levels determine to what extent the system will accept stiffness variations. This is an elegant approach

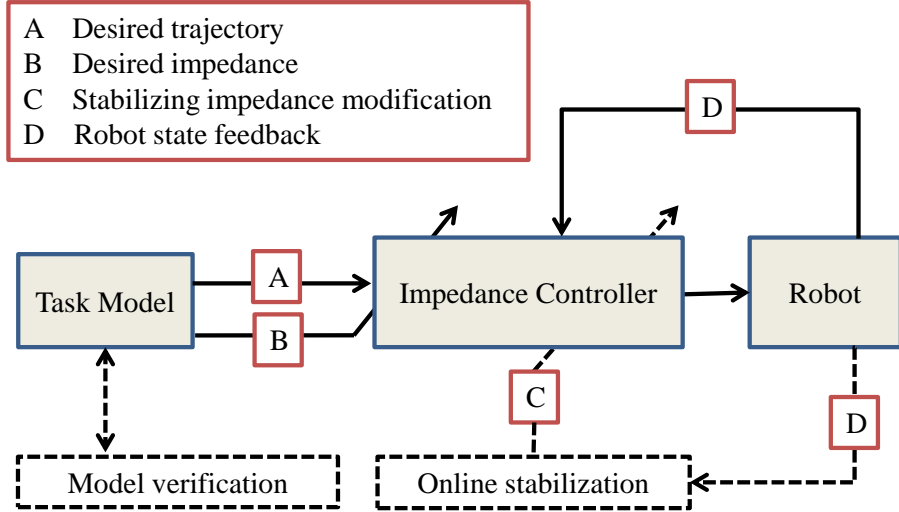


Figure 4.1: The block diagram illustrates two different approaches to guaranteed stability of varying impedance systems. Our proposed method consists of an state independent task model verification that can be carried out offline. The alternative approach involves online computation of an energy function and limitation of the impedance variations so as to ensure monotonic decline.

based on a sound energy-based idea. However, it has two important shortcomings: 1) It depends on the state of the robot, with the consequence that one can not guarantee beforehand the execution of a desired impedance profile 2) The performance depends strongly on the initial and threshold levels of energy in the tank¹. Our aim with this work is to provide a stability condition for varying stiffness and damping that is *state independent*. The most important practical advantage of such a constraint is that it can be verified offline, before execution of the task. Any standard impedance control architecture can subsequently be used for task execution, with a reassuring guarantee that the system can not go unstable. The two different approaches are illustrated in Fig. 4.1.

We focus on variable stiffness and damping, which are the impedance parameters that are most commonly varied for performance enhancement. We then propose a stability condition that relates the stiffness, damping and their rates of change. The constraint arises from the choice of a Lyapunov candidate function in which mixed position and velocity terms appear in the time derivative.

Related work for this chapter is reviewed in Section 2.1.2. This chapter is structured as follows. Section 4.2 provides some technical detail to the problem statement. In Section 4.3, we present our main result, a stability constraint for variable stiffness and damping control. The method is compared with the state-of-the art tank-based method (Ferraguti et al., 2013) two task simulations in Section 4.4.1. We then use our result to validate real impedance profiles in Section 4.4.2. We conclude with a discussion and outlook into future directions

¹For example, practically unstable behavior can emerge by setting the levels very high

of research in Section 4.5.

4.2 Problem Statement

The stability analysis in this Section applies to JS and CS controllers with dynamic decoupling described in Section 2.2.4. To keep the presentation general, the stability analysis will use the general notation from Section 2.2.4.

We assume the most common form of impedance control, which is to control the robot so that the following dynamic relationship is established between the generalized position error $\tilde{\boldsymbol{\xi}} \in \mathbb{R}^d$ and generalized force $\mathbf{F} \in \mathbb{R}^d$:

$$\mathbf{H}\ddot{\tilde{\boldsymbol{\xi}}} + \mathbf{D}\dot{\tilde{\boldsymbol{\xi}}} + \mathbf{K}\tilde{\boldsymbol{\xi}} = \mathbf{F} \quad (4.1)$$

Where $\mathbf{H} \in \mathbb{R}^{N \times N}$, $\mathbf{D} \in \mathbb{R}^{N \times N}$ and $\mathbf{K} \in \mathbb{R}^{N \times N}$ denote a desired inertia, damping and stiffness respectively. \mathbf{H} , \mathbf{D} and \mathbf{K} are positive definite and \mathbf{H} , \mathbf{K} are symmetric. The user-defined virtual inertia \mathbf{H} , damping \mathbf{D} and stiffness \mathbf{K} determine the behavior of the robot when subjected to external torque. If these parameters are constant, the system will be asymptotically stable for any symmetric positive definite choice of matrices \mathbf{H} , \mathbf{D} and \mathbf{K} . In this work, we are concerned with varying impedance control. Specifically, we will assume that \mathbf{H} remains constant while $\mathbf{D} = \mathbf{D}(t)$ and $\mathbf{K} = \mathbf{K}(t)$ are considered time varying functions.

To analyze the stability properties of Eq. (4.1), consider the standard Lyapunov candidate function consisting of kinetic energy with respect to the velocity error and the virtual potential energy stored by the stiffness:

$$V_1(\tilde{\boldsymbol{\xi}}, \dot{\tilde{\boldsymbol{\xi}}}, t) = \frac{1}{2}\dot{\tilde{\boldsymbol{\xi}}}^T \mathbf{H}\dot{\tilde{\boldsymbol{\xi}}} + \frac{1}{2}\tilde{\boldsymbol{\xi}}^T \mathbf{K}(t)\tilde{\boldsymbol{\xi}} \quad (4.2)$$

For compactness of notation, we will in the following refer to $V_1(\dot{\tilde{\boldsymbol{\xi}}}, \tilde{\boldsymbol{\xi}}, t)$ as simply V_1 . Differentiating V_1 along the trajectories of Eq. (4.1) with $\mathbf{F} = \mathbf{0}$ and \mathbf{H} constant yields:

$$\begin{aligned} \dot{V}_1 &= \dot{\tilde{\boldsymbol{\xi}}}^T \mathbf{H}\ddot{\tilde{\boldsymbol{\xi}}} + \dot{\tilde{\boldsymbol{\xi}}}^T \mathbf{K}(t)\dot{\tilde{\boldsymbol{\xi}}} + \frac{1}{2}\dot{\tilde{\boldsymbol{\xi}}}^T \dot{\mathbf{K}}(t)\tilde{\boldsymbol{\xi}} \\ &= \dot{\tilde{\boldsymbol{\xi}}}^T (-\mathbf{D}(t)\dot{\tilde{\boldsymbol{\xi}}} - \mathbf{K}(t)\tilde{\boldsymbol{\xi}}) + \dot{\tilde{\boldsymbol{\xi}}}^T \mathbf{K}(t)\dot{\tilde{\boldsymbol{\xi}}} + \frac{1}{2}\dot{\tilde{\boldsymbol{\xi}}}^T \dot{\mathbf{K}}(t)\tilde{\boldsymbol{\xi}} \end{aligned}$$

where symmetry of the stiffness matrix has been used. The mixed term with $\dot{\tilde{\boldsymbol{\xi}}}$ and $\tilde{\boldsymbol{\xi}}$ is hence cancelled but the potentially positive term with $\dot{\mathbf{K}}$ remains:

$$\dot{V}_1 = -\dot{\tilde{\boldsymbol{\xi}}}^T \mathbf{D}(t)\dot{\tilde{\boldsymbol{\xi}}} + \frac{1}{2}\dot{\tilde{\boldsymbol{\xi}}}^T \dot{\mathbf{K}}(t)\tilde{\boldsymbol{\xi}} \quad (4.3)$$

Eq. (4.3) is negative semidefinite for a negative semi-definite $\dot{\mathbf{K}}(t)$. Hence, we can conclude stability at the origin only if the stiffness is constant or decreasing. Assuming a $\tilde{\boldsymbol{\xi}} \neq 0$, increasing the stiffness entails an injection of potential energy

into the system, and it is hence intuitively clear that this practice can cause unstable behavior.

4.3 Ensuring Stability

As seen from Eq. (4.3), the classical energy function does not allow us to conclude stability with a varying stiffness profile. By inspection of the expression in Eq. (4.3), the obvious solution to the problem would be to design a controller that tries to follow the desired stiffness profile as well as possible, but limiting it when Eq. (4.3) becomes positive. It is soon realized, however, that such an approach has several disadvantages, the most important being 1) the admissible stiffness profile *depends on the state of the robot* and can hence not be known beforehand and 2) there are no guarantees in terms of capability of following the desired stiffness profile. By introducing the concept of an energy tank as in (Ferraguti et al., 2013), the second of these drawbacks can be somewhat relaxed.

4.3.1 STABILITY CONDITIONS ON STIFFNESS AND DAMPING PROFILES

Experience of varying stiffness control suggests that in general, reasonable varying stiffness profiles show no destabilization tendencies. This motivates the search for a less conservative Lyapunov candidate function than Eq. (4.2). In adaptive control, it is common to construct energy functions of weighted sums of the velocity error and the position error. The same approach can be used for varying stiffness control to establish state-independent stability conditions relating the stiffness and damping profiles. Consider the following Lyapunov candidate function:

$$V_2(\tilde{\boldsymbol{\xi}}, \dot{\tilde{\boldsymbol{\xi}}}, t) = \frac{(\dot{\tilde{\boldsymbol{\xi}}} + \alpha\tilde{\boldsymbol{\xi}})^T \mathbf{H}(\dot{\tilde{\boldsymbol{\xi}}} + \alpha\tilde{\boldsymbol{\xi}})}{2} + \frac{\tilde{\boldsymbol{\xi}}^T \beta(t)\tilde{\boldsymbol{\xi}}}{2} \quad (4.4)$$

where

$$\beta(t) = \mathbf{K}(t) + \alpha\mathbf{D}(t) - \alpha^2\mathbf{H} \quad (4.5)$$

with $\alpha > 0$ is some positive constant chosen such that $\beta(t) \geq 0$ for all $t > 0$. This candidate function is a generalized version of a Lyapunov function which is used for the analysis of time-varying scalar systems in (Slotine and Li, 1991). Note that $\alpha \rightarrow 0 \Rightarrow V_2 \rightarrow V_1$. In contrast to V_1 however, this function allows to establish sufficient constraints for stability that are *independent of the state*. This is formalized in the following theorem:

Theorem 4.1 (Stability conditions under dynamic decoupling). *Let \mathbf{H} be a constant, symmetric and positive definite matrix. Let $\mathbf{K}(t)$ and $\mathbf{D}(t)$ be symmetric, positive definite and continuously differentiable varying stiffness and damping profiles. Then, the system in Eq. (4.1) with $\mathbf{F} = \mathbf{0}$ is globally uniformly stable if there exists an $\alpha > 0$ such that $\forall t \geq 0$:*

1. $\alpha\mathbf{H} - \mathbf{D}(t)$ is negative semi-definite
2. $\dot{\mathbf{K}}(t) + \alpha\dot{\mathbf{D}}(t) - 2\alpha\mathbf{K}(t)$ is negative semi-definite

If in 2) semi-definiteness is replaced with definiteness, the stability property is in addition asymptotic.

Proof. The proof is given in Appendix B. □

It is perhaps not intuitively clear why the derivative of the damping appears in the second condition of Theorem 4.1. In the analysis in Section 4.2, $\dot{\mathbf{D}}$ does not appear in \dot{V}_1 (Eq. (4.3)). This means that for a constant stiffness, stability would be ensured by any positive definite \mathbf{D} , without any direct constraints² on $\dot{\mathbf{D}}$. Increasing the damping too fast however, can make the system converge at points with $\dot{\tilde{\xi}} = 0$ but $\tilde{\xi} \neq 0$ (Slotine and Li, 1991). The presence of $\dot{\mathbf{D}}$ in second condition in Theorem 4.1 prevents this from happening, since both $\dot{\mathbf{K}}$ and $\dot{\mathbf{D}}$ are in effect bounded by this constraint. Note that for robotic applications, this condition will typically be simplified to a form that has a more intuitive interpretation. We illustrate this point with two examples below, with a 1d-system given by:

$$m\ddot{q} + d(t)\dot{q} + k(t)q = 0 \quad (4.6)$$

Example 4.1 (Constant damping). Consider the system in Eq. (4.6) with constant damping $d(t) = d_0 > \alpha > 0$, implying $\dot{d}(t) = 0$. The stability conditions from Theorem 4.1 then reduce to:

$$d_0 > \alpha m \quad (4.6a)$$

$$\dot{k}(t) < 2\alpha k(t) \quad (4.6b)$$

Here, the second condition is an upper bound for the stiffness derivative that is proportional to the current stiffness and the level of damping.

Example 4.2 (Constant damping ratio). Consider the system in Eq. (4.6) with the damping chosen as $d(t) = 2\zeta\sqrt{mk(t)}$, where $\zeta > 0$ is a constant damping ratio.

Substituting $\dot{d}(t) = \frac{\zeta\sqrt{m}}{\sqrt{k(t)}}\dot{k}(t)$ into the second condition yields the following upper bound for the stiffness time-derivative:

$$\dot{k}(t) < \frac{2\alpha\sqrt{k(t)}^3}{\sqrt{k(t)} + \zeta\alpha\sqrt{m}} \quad (4.7)$$

4.3.2 VALIDATING IMPEDANCE PROFILES

The stability conditions presented in the previous section are constraints only

²There are constraints that follow from positive definiteness of \mathbf{D} .

on the stiffness and damping profiles. This means that it can be directly incorporated in an optimization or learning procedure, which is often utilized to generate impedance profiles. The outcome of these algorithms can hence be guaranteed to result in stable control.

In practice, the impedance parameter that has the most significant impact on task performance is the stiffness. Hence, it is often reasonable to give priority to the stiffness design. The damping can then be chosen to guarantee that the desired stiffness profile can be stably executed, typically by using critical damping. Inspection of the constraints of Theorem 4.1 reveals that the least conservative constraints are given by $\alpha\mathbf{H} - \mathbf{D}(t) = 0$. Hence, to have the least conservative constraints, α should be chosen as:

$$\alpha = \min_t \frac{\underline{\lambda}(\mathbf{D}(t))}{\bar{\lambda}(\mathbf{H})} \quad (4.8)$$

where $\bar{\lambda}(\cdot)$ and $\underline{\lambda}(\cdot)$ denote the largest and smallest eigenvalue respectively. With this choice, the first condition of Theorem 1 is by construction satisfied for all t . Hence, what remains to be verified is the second condition of Theorem 4.1. A simple verification procedure is as follows:

1. Given some ideal desired $\mathbf{D}(t)$ and \mathbf{H} , determine α according to Eq. (4.8).
2. Verify that $\dot{\mathbf{K}}(t) + \alpha\dot{\mathbf{D}}(t) - 2\alpha\mathbf{K}(t)$ is negative semi-definite for all $t > 0$.
3. If not verified, modify stiffness and/or damping profile³.

Below, we show how the constraints in Theorem 4.1 relate to the eigenvalues of the matrices in the general case. Consider $A = -\mathbf{D} + \alpha\mathbf{H}$. This matrix needs to be negative definite for negative definiteness of \dot{V}_2 . \mathbf{H} and \mathbf{D} are symmetric, which implies A is symmetric and its negative definiteness is hence equivalent all its eigenvalues being negative. In particular, its largest eigenvalue $\bar{\lambda}(A)$ should be negative.

$$\bar{\lambda}(A) = \sup_{\|v\|=1} v^T A v \leq \underbrace{\sup_{\|v\|=1} v^T (-\mathbf{D}) v}_{=\bar{\lambda}(-\mathbf{D})} + \underbrace{\sup_{\|v\|=1} v^T (\alpha\mathbf{H}) v}_{\alpha\bar{\lambda}(\mathbf{H})} \quad (4.9)$$

Above, the triangle inequality for the supremum norm has been used. Thus, we have:

$$\bar{\lambda}(A) \leq -\bar{\lambda}(\mathbf{D}) + \alpha\bar{\lambda}(\mathbf{H}) \quad (4.10)$$

since $\bar{\lambda}(-\mathbf{D}) = -\bar{\lambda}(\mathbf{D})$. Consequently, we have that negative definiteness of A is implied by the following bound on the smallest eigenvalue of \mathbf{D} :

$$\bar{\lambda}(\mathbf{D}) > \alpha\bar{\lambda}(\mathbf{H}) \quad (4.11)$$

³E.g. increase constant damping or increase the damping ratio for the common case that the damping is varying with the stiffness to ensure a constant damping ratio.

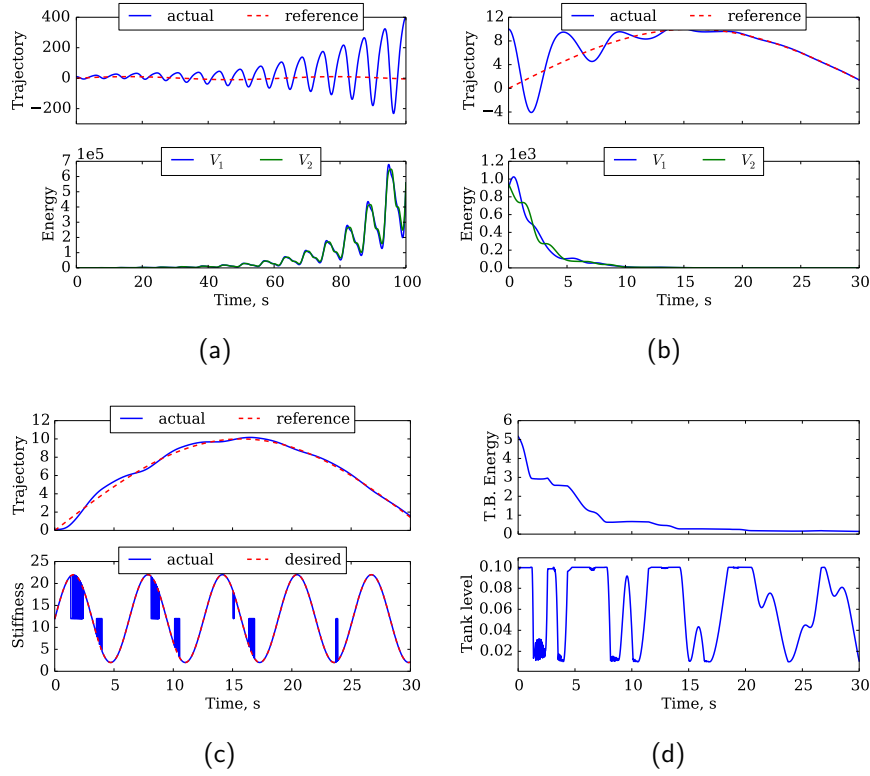


Figure 4.2: **a)** Trajectory and the Lyapunov candidates V_1 and V_2 during a simulation of a single d.o.f with mass 10 Kg, damping 1 Ns/m and desired trajectory and stiffness given by Equations (4.15a) and (4.15b) respectively. **b)** The same system with modified impedance profile that is validated according to procedure in Section 4.3.2. **c)** Trajectory and stiffness during the first 30 seconds simulated with the tank-based controller from (Ferraguti et al., 2013). **d)** The tank-based energy function and the tank level during the 10 seconds of simulation.

Hence, α defines a lower bound for the minimum eigenvalue of \mathbf{D} as a multiple of the maximum eigenvalue of \mathbf{H} .

Now consider the second condition, which bounds the rate of change in stiffness. The following matrix should be negative definite:

$$C = \dot{\mathbf{K}} + \alpha \dot{\mathbf{D}} - 2\alpha \mathbf{K} \quad (4.12)$$

As above, we use the triangle inequality to bound the largest eigenvalue of C :

$$\bar{\lambda}(C) \leq \bar{\lambda}(\dot{\mathbf{K}}) + \alpha \bar{\lambda}(\dot{\mathbf{D}}) - 2\alpha \underline{\lambda}(\mathbf{K}) \quad (4.13)$$

It follows that a sufficient condition for negative definiteness of C is given by:

$$\bar{\lambda}(\dot{\mathbf{K}}) < 2\alpha \underline{\lambda}(\mathbf{K}) - \alpha \bar{\lambda}(\dot{\mathbf{D}}) \quad (4.14)$$

4.4 Evaluation

4.4.1 SIMULATIONS

To illustrate the properties of the proposed approach, we present a set of simulations comparing the stability analysis tools from Section 4.3 with the recently proposed tank-based approach to varying stiffness control (Ferraguti et al., 2013).

TRACKING TASK

First we shall consider an example of 1 dof system that is unstable due to a varying stiffness profile. This is a reproduction of the simulation in (Ferraguti et al., 2013). The reference trajectory and the desired stiffness profile are given by:

$$x_t^d = 10\sin(0.1t) \quad (4.15a)$$

$$k_t^d = k_0 + 10\sin(0.1t) \quad (4.15b)$$

with $k_0 = 12$. The system is simulated with a mass of 10 Kg and a damping of 1 Ns/m. Fig. 4.2a shows a simulation of 100 seconds of this system. As was shown in (Ferraguti et al., 2013), the tank-based method can stabilize the system. Figures 4.2c and 4.2d show the results from the first 10 seconds of simulation with the tank-based controller. As can be seen, when the minimum tank level is reached, there are discontinuities in the stiffness profile. This effect is a direct result of the formulation, which will fall back to a predefined constant component of stiffness when the tank is empty. When and to what extent this effect occurs depends on the choice of the open tank parameters and more importantly the state of the robot as the stiffness variations take place.

As can be seen in the energy plot in Fig. 4.2a, V_1 as well as V_2 are non-monotonic and increasing. According to the design procedure described in Section 4.3.2, the impedance profile should be modified so that $\dot{k} - 2\alpha k$ remains negative. To achieve this, the damping and base stiffness were increased to $d = 4$ and $k_0 = 18$ respectively. Fig. 4.2b shows the results of simulating this modified system during 100 seconds. As expected, the system is now stable. Note that the classical energy function, V_1 , is still locally increasing, which depending on the chosen tank parameters would still require limitation of the stiffness profile with the tank-based controller.

REGULATION WITH PERTURBATIONS

The advantage of ensuring stability of a varying impedance profile without dependency on state measurements is most clear in situations where perturbations are to be expected during task executions and a reliable trajectory prediction

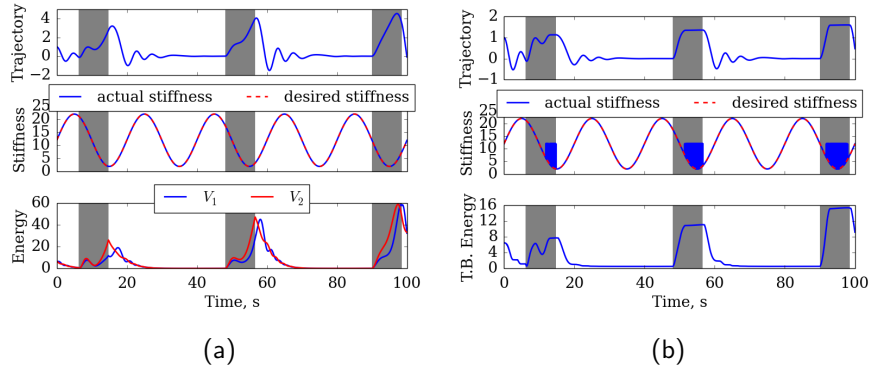


Figure 4.3: **a)** Trajectory, stiffness and evolution of the classical Lyapunov function as well as the one used in this work. The data comes from a 100 s simulation of a regulation task. The system is perturbed by the application of a constant force during the shaded areas. Here, the standard impedance controller is used since the desired stiffness profile is guaranteed to be stable. This fact is confirmed here by the monotonic decrease of V_2 outside the shaded regions. **b)** Same as a) but system controlled with tank-based approach from (Ferraguti et al., 2013). The bottom plot shows the time series of the tank-based energy function. Note that it is in particular during perturbations that the system is unable to use the desired stiffness.

is not possible. By inspection of Eq. (4.3), it is clear that it is particularly in the presence of a large position error $\tilde{\xi}$ that energy is injected into the system if the stiffness is increased. Any method relying on observing this energy function by state measurements can hence be expected to be significantly active during perturbations. To illustrate this, we simulated a one d.o.f regulation task with a varying stiffness profile given by:

$$k = 12 + 10 \sin\left(\frac{\pi}{10}t\right) \quad (4.16)$$

and a constant damping 4 Ns/m and a mass of 10 Kg. Fig. 4.3a shows that the impedance profile is validated by Theorem 4.1 and the trajectory resulting from the simulation. A constant, positive perturbation force was applied in intervals shaded gray in the plots. Fig. 4.3b shows the results of the tank-based method. Note that it is in particular when the system is perturbed that the tank controller has to fall back to the constant stiffness value. We wish to emphasize that this is not a problem particularly related to the tank-based stabilization method, but rather a general problem of any stabilization method relying on state measurements during task execution.

4.4.2 VALIDATION OF REAL IMPEDANCE PROFILES

Although it has been shown that varying stiffness can lead to unstable behavior, experience shows that this is a rare occurrence in practice. There exists already

several works that have used varying stiffness controllers which have not resulted in unstable behavior. However, stability has only been proven in special cases where the stiffness is varied according to an adaptive control law such as (Yang et al., 2011) or with a particular parametric model as in our previous work (Khansari-Zadeh et al., 2014). For the more general case, were the stiffness can be considered a time varying function just like the trajectory, Theorem 4.1 can be used to validate or reject stiffness profiles. Since our stability condition is conservative⁴, it is important to investigate if in practice it can be used to validate stiffness trajectories. Therefore, we gathered data from three previous works by ourselves and others, and tested to see if our stability condition could validate these impedance profiles. In each case, a control scheme with inverse dynamics and damping varying as a function of the stiffness so as to ensure a critically damped system was assumed. First, for each case a value of α was selected according to Eq. (4.8). Then, the following time-varying matrix was computed:

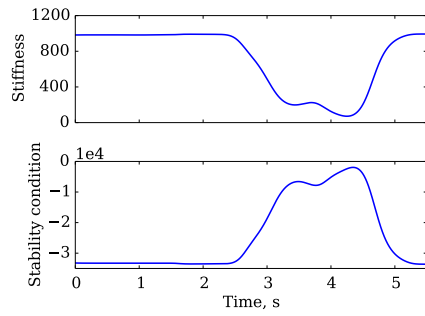
$$\dot{\mathbf{K}}(t) + \alpha\dot{\mathbf{D}}(t) - 2\alpha\mathbf{K}(t) \quad (4.17)$$

The stability according to Theorem 4.1 is then confirmed if all eigenvalues of Eq. (4.17) remain negative. The first data set is a match-lighting task from Section 3.3.3. This experiment was carried out on the KUKA LWR, see Fig. 4.4b. A stiffness profile was taught to the robot using a physical human-robot interface designed for this purpose. The resulting stiffness profile is a constant, high stiffness for all joints except the elbow joint (encircled in Fig. 4.4b), in which the stiffness is reduced as the robot strikes the match on the matchbox. The stiffness profile for the elbow joint is shown in Fig. 4.4a, top. Fig. 4.4a, bottom shows the evolution of the largest eigenvalue of Eq. (4.17). Note that only one of the eigenvalues are plotted since the other 6 remain constant throughout the task.

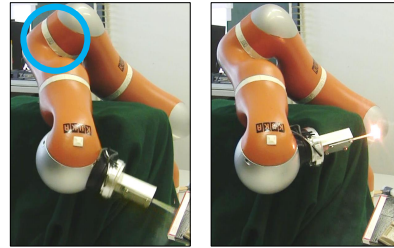
The second data set comes from Calinon et al. (Calinon et al., 2014). The task is a 2D reaching task where the final part of the trajectory passes through a narrow path leading to the target. The trajectory and varying stiffness are generated online by Gaussian Mixture Regression (GMR) and minimum intervention control scheme using the covariance from GMR to find a stiffness profile. Fig. 4.4c shows the eigenvalues of the stiffness matrix and Eq. (4.17). As can be seen, both eigenvalues remain negative throughout the time series and the impedance profile is hence validated as stable.

The third data set comes from Buchli et al. (Buchli et al., 2011), who applied the reinforcement learning algorithm PI2 to simultaneously learn the trajectory and stiffness profile repeated task trials evaluated by a given cost function. The task is a via-point task in a 6d joint space implemented on a KUKA LWR robot (the last joint was ignored). The cost function was designed to favor a compliant behavior when accuracy is not needed. This resulted in a stiffness profile with

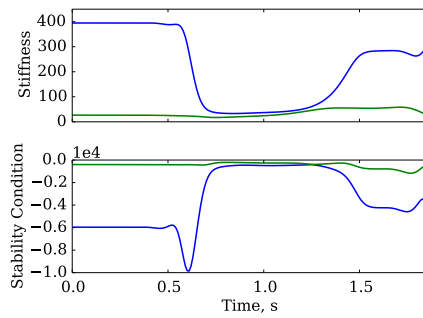
⁴In the sense that there are stable impedance profiles that do not satisfy Theorem 4.1.



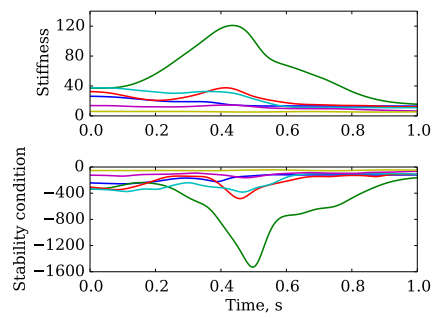
(a)



(b)



(c)



(d)

Figure 4.4: **a)** Stiffness profile and evolution of stability condition given by for the match-lighting task. **Bottom:** The evolution of the maximum eigenvalue of Eq. 22

a low stiffness that locally increases when the robot passes near the via point. Fig. 4.4d shows the time series of the eigenvalues of the stiffness matrix and Eq. (4.17) respectively. As in the previous examples, all eigenvalues clearly remain negative.

4.5 Discussion and conclusion

The most fundamental property that should be required from a control system is stability. This is especially important in applications of robots near humans, in which variable impedance control is particularly interesting to use. While experience tells us that unstable behavior arising from variable impedance is quite rare in practice, it is important to understand this issue better and it is crucial to subject the impedance variations to constraints so that stability is guaranteed. We have proposed a novel, state-independent stability condition for varying stiffness and damping profiles.

Impedance control commonly used in situations where significant departure from the reference trajectory can be expected by either temporary perturbations or physical constraints in the environment. For this reason, the reference trajectory is often renamed virtual trajectory in impedance control literature, highlighting the fact that perfect tracking of the reference may not be possible or even desired. As shown by the simulation in Section 4.4.1, using the classical Lyapunov function as a stability observer will make it impossible to increase the stiffness in the presence of significant position errors. A clear advantage must be ascribed to the proposed method (Section 4.3) in this regard.

The main limitation of the proposed approach is that it requires the inertia matrix \mathbf{H} to be constant. This in turn requires model knowledge and dynamic decoupling as described in Section 2.2.4. Our method is not applicable to the simpler impedance controllers that use the inherent inertia of the robot, since the latter is configuration dependent and hence not constant when the robot is moving. The state-dependent tank based method proposed in (Ferraguti et al., 2013) does not have this limitation.

It is possible to construct variable impedance profiles that yield qualitatively stable behavior but that are not validated by Theorem 4.1. This is because Theorem 4.1 is a conservative guarantee of stability. Our experience is that it will validate reasonably chosen impedance profiles, as shown in Section 4.4.2.

The control architecture in Chapter 3 and in this chapter has been focused on a time dependent reference trajectory and a varying impedance profile. This gives ample flexibility for many tasks, but not all. One very important class of tasks which can generally not be solved using this architecture is insertion tasks. We address this in the next chapter by introducing online generation of the reference trajectory based on sensed interaction forces.

HAPTIC REFERENCE ADAPTATION FOR INSERTION TASKS

5.1 Introduction

In Chapter 3, we assumed that we had at our disposal a reference trajectory and focused on how acquire a varying impedance in a task specific manner. Insertion tasks constitute an interesting class of tasks which generally requires active adaptation of the reference trajectory in response to sensed forces (Asada, 1993), which was not present in Chapters 3 and 4. The difficulty of the peg-in-hole task and its importance for assembly operations in the manufacturing industry are well documented in literature Hannaford et al. (1991). If the pose of the parts can be tracked with negligible inaccuracy, the problem can be solved by generating trajectories based on the geometry of the parts Lozano-Perez et al. (1984). This method fails in the common case where the uncertainty of the pose estimate or the inaccuracy of control is significantly larger than the clearance. Humans can carry out the peg-in-hole task with or without visual feedback by relying on haptic feedback for completing the task. By feeling the interaction between the peg and the hole, humans are able to instantly adapt the pose of the peg to complete the task. It would be highly desirable to transfer this skill to robots. In this chapter, we take a step in this direction and propose a methodology for transferring task-specific insertion-skills to robots. The approach is based on LfD and uses data recorded from *collaborative task executions*, see Fig. 5.5, to learn a model allowing to adapt the reference orientation based on the sensed wrench. Some background and a review of related works is given in Section 2.1.5.

5.2 Problem Statement

As mentioned in the introduction, the peg-in-hole problem in which the clearance is larger than the uncertainty of the precision of position sensing and control can be solved using traditional kinematic planning and position or velocity control. We consider the more challenging case where the uncertainty in position sensing and control is significantly larger than the clearance.

Assuming that the peg is grasped by the robot, the peg-in-hole task can be decomposed as illustrated in Fig. 5.2. During the development of this work, we

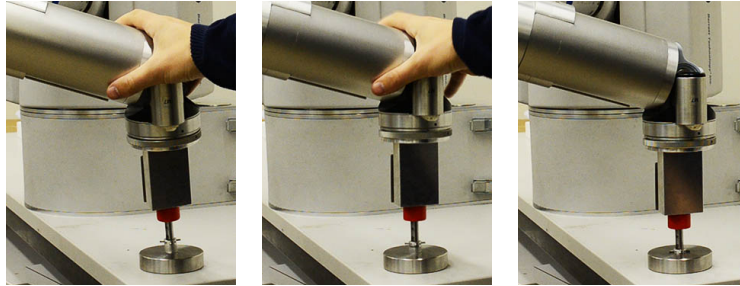


Figure 5.1: **Left:** The robot has found the hole but the peg is jammed due to incorrect orientation. The teacher grasps the robot to correct the orientation. **Middle:** The teacher has corrected the orientation. **Right:** As the teacher releases the robot, it automatically completes the insertion.

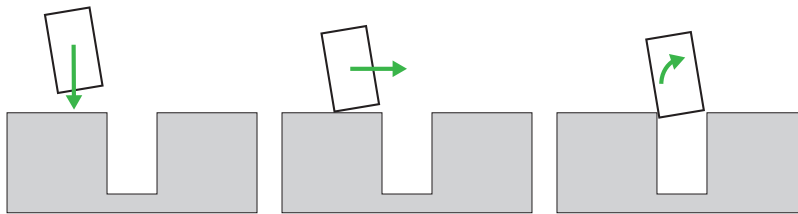


Figure 5.2: The three phases of the insertion problem illustrated in 2D. **Left:** Moving towards the surface on which the hole is located. **Middle:** Searching for the hole along the surface. **Right:** Insertion phase consisting of guiding the head of the peg into the hole and aligning the orientation with the hole.

found that the first two steps, i.e. finding the hole surface and the hole position can be completed with simple sequencing as described in Section 5.5. In this chapter, we focus on the last (and technically most challenging) part: correcting the orientation of the peg such that its head falls completely into the hole, and subsequent realignment of the peg to the natural constraints of the hole. If the clearance is tight, even a slight misalignment gives rise to jamming in this phase.

We posit that for peg-in-hole insertion, the sensed wrench is sufficient information to guide the insertion of the peg. Our goal with this work is to use the human reactivity in the insertion phase as a model for developing a non-linear model that can be used by the robot to successfully perform the task on its own. This model will monitor the sensed contact wrench $\phi \in \mathbb{R}^6$ and select an appropriate angular velocity $\omega \in \mathbb{R}^3$ depending on ϕ .

5.3 Learning Haptic Reference Adaptation from Human Demonstrations

As discussed in Section 5.2, we focus on the orientation alignment phase of the peg-in-hole task. In the orientation alignment phase, we posit that sensing the contact wrench provides sufficient information for realigning the peg until insertion is completed. This corresponds to choosing an angular velocity for the reference trajectory that will move the peg such that it slides into the hole. Humans clearly perform the insertion with ease, with the peg in hand or by guiding the robot kinesthetically. By measuring the sensed wrench and corresponding angular velocity when a teacher is helping the robot to complete the insertion, we can acquire a training data set $\{\phi_n, \omega_n\}_{n=1}^N$ consisting of N training points with a sensed wrench and corresponding corrective velocity.

Our approach uses a GMM (described in Section 2.2.5) to represent the joint probability distribution of the sensed wrench and the corrective angular velocity using a limited set of parameters. The model is trained offline, using the standard EM algorithm (Dempster et al., 1977).

It is straightforward to condition a GMM on part of the vector whose probability distribution it encodes. The GMR procedure involves conditioning the GMM on an input vector (in this chapter the sensed wrench), and using the mean of the resulting distribution — which is again Gaussian — as an estimate for the corresponding output (in this chapter the corrective angular velocity). The conditional distribution can also be used for sampling, resulting in a random search which is guided by the sensed wrench. As will be shown later, this procedure proved better suited than GMR for the peg-in-hole task. We omit details of the GMM equations here (refer to Section 2.2.5), but recall that the GMM can be conditioned on the sensed wrench, resulting in a distribution over angular velocities:

$$p(\boldsymbol{\omega}|\boldsymbol{\phi}) = \sum_{k=1}^K p(k|\boldsymbol{\phi})p^k(\boldsymbol{\omega}|\boldsymbol{\phi}) \quad (5.1)$$

where $p(k|\boldsymbol{\phi})$ is defined by Eq. (2.30) and where the local conditioned models $p^k(\boldsymbol{\phi}|\boldsymbol{\omega})$ is given by Eq. (2.29), in each case replacing $\boldsymbol{\xi}_{\mathcal{I}}$ and $\boldsymbol{\xi}_{\mathcal{O}}$ by the sensed wrench $\boldsymbol{\phi}$ and the angular velocity $\boldsymbol{\omega}$ respectively. From this distribution, one can acquire a functional relationship from $\boldsymbol{\phi}$ to $\boldsymbol{\omega}$ by taking the expectation of the conditional:

$$\hat{\boldsymbol{\omega}} = \text{E}\{p(\boldsymbol{\omega}|\boldsymbol{\phi})\} = \sum_{k=1}^K p(k|\boldsymbol{\phi})\mathbf{m}_{\boldsymbol{\omega}|\boldsymbol{\phi}}^k \quad (5.2)$$

where $\mathbf{m}_{\boldsymbol{\omega}|\boldsymbol{\phi}}^k$ is given by Eq. (2.29b).

5.4 Trajectory Generation and Robot Control

When the robot has initialized the final stage of the insertion, the peg has partially fallen into the hole and there is an interaction wrench between the surface and the peg. A reference orientation is the generated by integrating the reference orientation at a discrete rate of $1/T$, which is a predetermined update rate. Let \mathbf{R}_k denote the reference orientation at one such update time. The reference next reference orientation \mathbf{R}_{k+1} is then updated by the corrective angular velocity $\boldsymbol{\omega}_k$ as:

1. Find the rotation corresponding to applying $\boldsymbol{\omega}_k$ constant during the update period T :

$$\boldsymbol{\psi}_{k+1} = T\boldsymbol{\omega}_k \quad (5.3)$$

2. Find the corresponding rotation matrix $\Delta\mathbf{R}_{k+1}$ using the standard Rodriguez rotation formula:

$$\Delta\mathbf{R}_{k+1} = \cos(a)\mathbf{I}_{3 \times 3} + \sin(a)[\mathbf{b}]_{\times} + (1 - \cos(a))\mathbf{b} \otimes \mathbf{b} \quad (5.4)$$

with $a = \|\boldsymbol{\psi}_{k+1}\|$ and $\mathbf{b} = \frac{\boldsymbol{\psi}_{k+1}}{a}$ and where $[\cdot]_{\times}$ and \otimes denote the cross-product matrix and tensor product respectively.

3. Update the reference orientation:

$$\mathbf{R}_{k+1} = \Delta\mathbf{R}_{k+1}\mathbf{R}_k \quad (5.5)$$

The reference orientation is initialized with the actual orientation at the beginning of the insertion phase. The reference orientation is hence piece-wise constant during the period T . Note that the above procedure is not numerical integration of the angular velocity — rather it should be seen as a series of rotations around piece-wise fixed rotation axes.

Our approach for deriving a reactive behavior mapping sensed wrench to

corrective angular velocity is based on noisy training data. It is hence unavoidable — regardless if sampling or regression is used — that incorrect angular velocities will sometimes be chosen. To avoid hazardous build up of contact forces when incorrect velocities are applied, it is therefore necessary to use a compliant controller for executing the reference trajectory resulting from applying the corrective angular velocity. To this end, we use the Cartesian impedance controller without inertia shaping described in Section 2.2.4.

5.5 Experimental Setup

To evaluate the proposed system, several experiments with different adaptation schemes were carried out on a 7-degree-of-freedom Barrett WAM for inserting a peg into a hole. The primary setup consisted of a steel peg with a diameter of 15.84 mm and a steel hole with a diameter of 15.98 mm. A secondary setup consisting of a wooden peg with diameter 16.5 mm and a rubber hole with diameter 18 mm was used for evaluating the generalization capabilities of the proposed approach.

5.5.1 PREPARING FOR INSERTION

As discussed in Section 5.2, we found that approaching the hole surface and finding the position of the hole was easy to achieve through task sequencing using transients in the wrist mounted force-sensor to indicate partial goal achievement. The procedure can be summarized as follows:

1. Place the peg above the hole surface.
2. Move vertically until a transient is detected on the z-axis of the force sensor, see Fig. 5.3 left.
3. Move in an expanding spiral until a transient in the x-y plane of the force sensor is detected, see Fig. 5.3 middle.

These motion patterns were fully hand-coded, since we focus on the last phase of the insertion problem.

5.5.2 DATA COLLECTION AND MODEL LEARNING

Before each demonstration, a random reference orientation $\mathbf{R}^r = \mathbf{R}^f \mathbf{R}^h$ was chosen, with a perturbation by the rotation \mathbf{R}^f of the true orientation of the hole \mathbf{R}^h . \mathbf{R}^f is a uniformly distributed random rotation with angles around x and y axes in $[-20^\circ, 20^\circ]$. With this reference orientation, the robot then prepared for insertion as described in Section 5.5.1.

Once the robot had found the hole, the stiffness was reduced to $k^p = 100$ N/m and $k^o = 0.5$ Nm/rad, enabling the user to move the robot with little effort

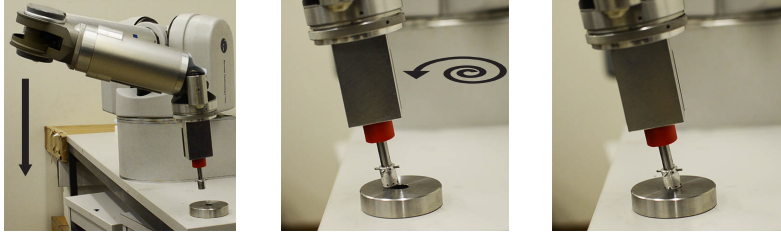


Figure 5.3: **Left:** The robot moving vertically towards the surface until contact is detected. **Middle:** An expanding spiral is used to search for the hole on the surface. **Right:** The robot stops searching when an increase of the measured force in the horizontal plane is detected. At this stage, some part of the peg has fallen into the hole and the insertion begins.

Joint angles	Sensed wrench	Cart. pos.	Cart. orient.	Reference Cart. pos./orient.	Kinematic Jacobian	Cart. error	Joint torque
q	ϕ	p	R	p^r/R^r	J	\tilde{x}	τ

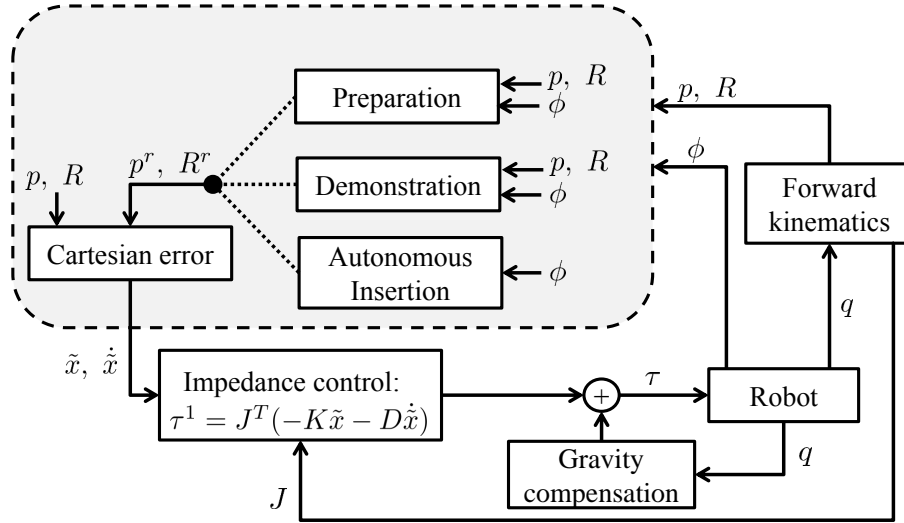


Figure 5.4: Block diagram of the experimental setup. The robot is controlled with Cartesian impedance control. The reference position and orientation are computed according to the sequencing described in Section 5.5.1 for the preparatory step, constant when collecting demonstrations (Section 5.5.2) and updated as described in Section 5.5.3 during autonomous task execution. The Cartesian impedance controller and error computation are described in Section 2.2.4.

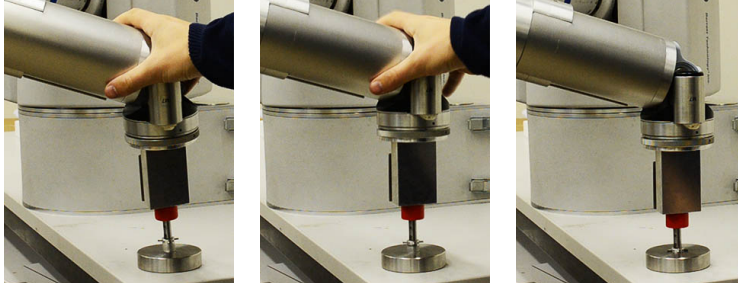


Figure 5.5: Left: The robot has found the hole but the peg is jammed due to incorrect orientation. The teacher grasps the robot to correct the orientation. Middle: The teacher has corrected the orientation. Right: As the teacher releases the robot, it automatically completes the insertion.

during the demonstrations. The teacher performed demonstrations by helping the robot to insert the peg by physically guiding it, see Fig. 5.5. During this time, the sensed wrench and the corrective angular velocity were continuously recorded. 8 corrective demonstrations were given from different initial rotation error configurations, yielding a total of 2478 data points.

GMMs with numbers of Gaussians ranging from 1 to 20 were then trained using EM on the collected data set. Evaluation using BIC on the trained models favored a GMM with 6 components, which was therefore selected as the model for experimental evaluation.

To compare our approach with random search that does not take the sensed wrench into account, we used two controllers, sampling either from a Gaussian or uniform distribution. To make the comparison fair, we ensured that the two random search controllers were somewhat informed about the task by adapting their parameters according to the observed data. The mean of the Gaussian was chosen to $\boldsymbol{\mu}_\omega^N = \mathbf{0}$ and the covariance matrix $\boldsymbol{\Sigma}_\omega^N$ was computed as the empirical covariance matrix of the collected samples of angular velocities.

$$\boldsymbol{\Sigma}_\omega^N = \frac{1}{N} \sum_{i=1}^N (\boldsymbol{\omega}_i - \mathbf{c})(\boldsymbol{\omega}_i - \mathbf{c})^T \quad (5.6)$$

where \mathbf{c} denotes the empirical mean of the angular velocities in the training set:

$$\mathbf{c} = \frac{1}{N} \sum_{i=1}^N \boldsymbol{\omega}_i$$

The uniform distribution was independent in each component $j = 1, 2, 3$ of $\boldsymbol{\omega} = [\omega^1, \omega^2, \omega^3]^T$. Bounds a^j and b^j for each component ω^j were selected as the minimum and maximum value of that component in the centered training data. We denote the joint distribution of the three components by:

$$\mathcal{U}(\mathbf{a}, \mathbf{b}) = \mathcal{U}(a^1, b^1)\mathcal{U}(a^2, b^2)\mathcal{U}(a^3, b^3). \quad (5.7)$$

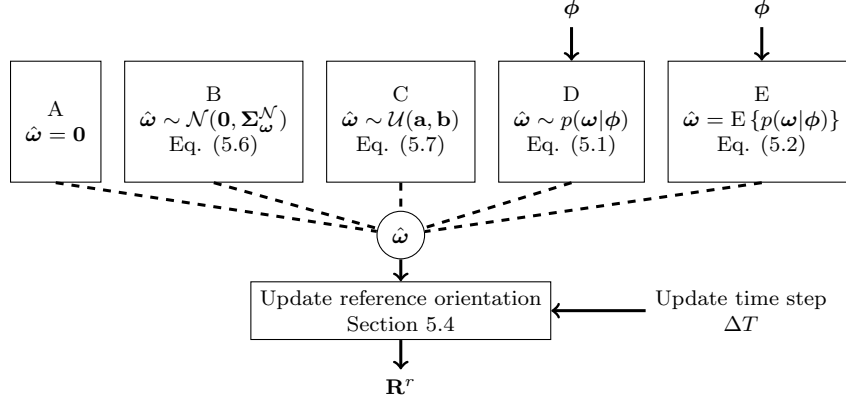


Figure 5.6: Block diagram detailing the autonomous task execution. The different reference adaptation schemes A-E are described in Section 5.5.3.

5.5.3 AUTONOMOUS TASK EXECUTION

Autonomous task execution for evaluating the performance of the proposed system was preceded by preparation for insertion as described in Section 5.5.1. For insertion, the controller maintained its reference position \mathbf{p}^r , while the reference orientation \mathbf{R}^r was adapted with the angular velocity $\hat{\omega}$ as described in Section 5.4. An update period of 10ms was used. Constant diagonal stiffness and damping matrices were used. The translational stiffness was chosen identical for the three axes to $k^p = 400$ N/m. The rotational stiffness was chosen identical around the three axes to $k^o = 2$ Nm/rad. The damping was designed based on the stiffness as described in Section 2.2.4. The stiffness value was chosen empirically to limit the contact forces to a safe range. Five different setups were used, which differ in the way the angular velocity is computed:

- A. No orientation correction, $\hat{\omega} = 0$.
- B. Random orientation correction not considering sensed wrench, Gaussian distribution $\hat{\omega} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\hat{\omega}}^{\mathcal{N}})$, refer to Eq. (5.6).
- C. Random orientation correction not considering sensed wrench, uniform distribution, refer to Eq. (5.7).
- D. Orientation correction drawn from conditional derived from the GMM, Eq. (5.1).
- E. Orientation correction acquired through GMR, Eq. (5.2).

A block diagram describing the update for the different adaptation schemes can be seen in Fig. 5.6.

5.6 Results

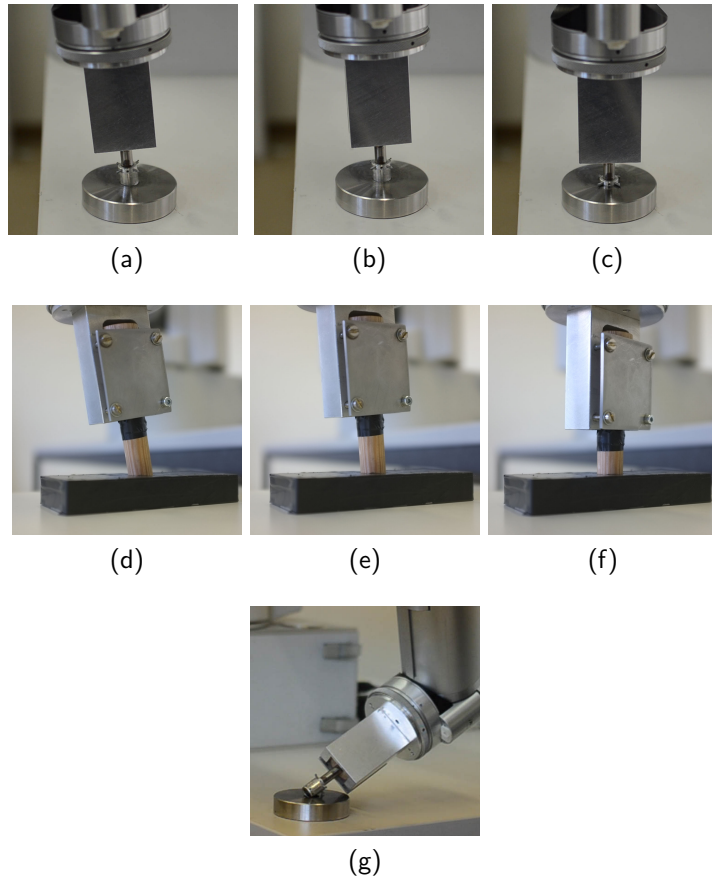


Figure 5.7: **a-c)** Autonomous insertion of the steel peg and hole. **d-f)** Autonomous execution using the model learned on the steel peg and hole for insertion of the wooden peg into a rubber hole.

Two sets of experiments were carried out. The first one aims at comparing the adaptation controllers A-E on the same peg and hole. The second set of experiments was carried out to investigate the generalization capabilities of the best performing control scheme.

5.6.1 COMPARISON OF ADAPTATION SCHEMES

To compare the different adaptation schemes, 50 insertion attempts with the steel peg and hole (see Fig. 5.7a) were carried out for each controller A-E. For each trial, an initial reference orientation was chosen from a grid of equally spaced orientations with rotations from the true hole orientation around the x- and y-axes bounded in $[-20^\circ, 20^\circ]$.

Several aspects can be considered for evaluating a peg-in-hole insertion. The first aspect is the rate of success. It was observed that if the insertion had not been completed within 20 seconds, the robot was typically locked in a static position. Therefore, a trial was considered successful if it fully inserted the peg in less than 20 seconds. Another point of comparison is speed of insertion, which was computed as the average completion time for successful results. In many possible applications for peg-in-hole, e.g. assembly of fragile parts, it is important that the contact wrench is kept low during insertion. Therefore, the amplitude of the contact force and torque during insertion were also used as points for comparison.

The results from 50 trials of each of the reference adaptation methods A-E is given in Table 5.1. Unsurprisingly, the worst performing controller is A, which does not adapt the orientation. This controller only succeeds when the initial reference orientation (refer to Section 5.5.3 for details) is close enough to the true orientation of the hole. Note that the completion time for this controller is much lower than for the others since the only successful trials are those in which the peg slides directly into the hole without correction.

Controller B is randomly sampling angular velocities from a Gaussian distribution as described in Section 5.5.3. This distribution is visualized in Fig. 5.8a. The random search can only aid the insertion if an angular velocity of the correct direction is sampled. Naturally, the majority of the sampled velocities are incorrect, leading to low rate of success and high completion time as seen in Table 5.1. Controller C samples from a uniform distribution and achieves a higher rate of success than B. This could possibly be explained by the fact that the samples in C are more dispersed from zero than the samples in B. A common outcome of the trials in both B and C were a sideways collapse of the peg as seen in Fig. 5.7g due to an incorrect angular velocity with a high speed in the wrong direction being chosen. In these cases, the peg has left the hole and the task can no longer be completed through orientation adaptation.

Controller D is sampling angular velocities from a distribution that depends

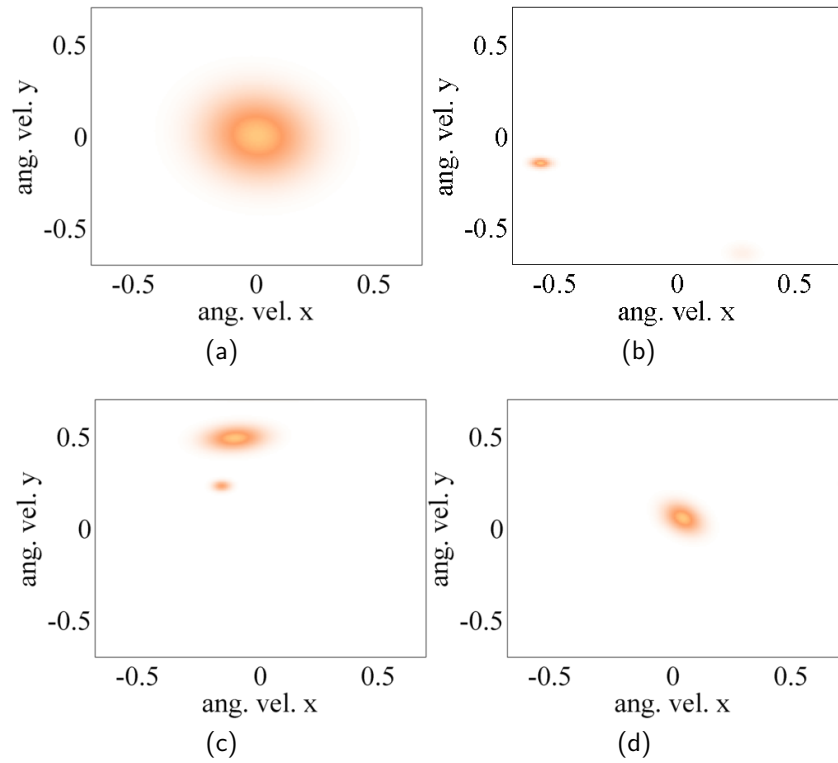


Figure 5.8: Probability densities for corrective angular velocity around the x- and y-axes. **a)** The figure shows the Gaussian distribution described in Eq. (5.6), which does not take into account the sensed wrench. **b,c, and d)** Examples of conditioning the learned GMM on three different instances of sensed wrench.

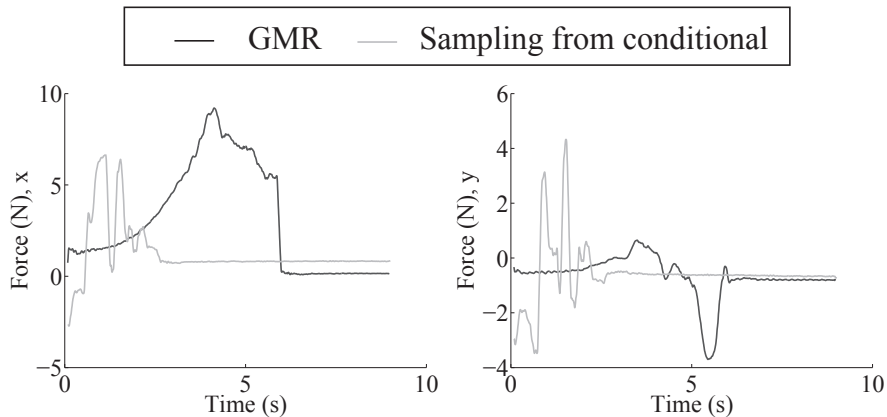


Figure 5.9: The figures show the contact force during an example of a successful insertion using adaptation with GMR (controller D) and sampling from the conditional of the learned model (controller E).

Table 5.1: Experimental results based on 50 trials for each of the five reference adaptation methods described in Section 5.5.3. The initial reference orientation was chosen from a grid as described in Section 5.5.3. Statistics are given as: mean(standard deviation).

	Rate of success, %	Completion time, s	Norm of contact force, N	Norm of contact torque, Nm
A	8	0.5 (0.7)	1.4 (0.7)	0.19 (0.15)
B	12	9.7 (7.0)	2.9 (1.4)	0.35 (0.24)
C	22	12.7 (4.2)	2.7 (1.5)	0.35 (0.21)
D	72	8.7 (5.6)	4.4 (2.6)	0.55 (0.36)
E	36	9.3 (5.1)	4.2 (1.9)	0.50 (0.23)

on the sensed wrench. This distribution is very different from the one used by controllers B and C, as can be seen in Fig. 5.8. Importantly, the mean of the conditional is in most cases far away from the origin, reflecting the fact that the human teacher generally did not ignore the presence of a contact wrench in the demonstration data. The higher completion time compared to controllers A, B and C is due to the fact that controller D can complete insertion in difficult cases that fail (and are hence not considered for the average completion time) with A,B or C.

Controller E uses GMR to choose an angular velocity based on the sensed wrench. We observed during the trials that this controller often gets stuck in static jamming conditions that it is unable to resolve. This is explained by biased noise in the wrench sensing and by errors in the trained model. Controller D does not suffer from this effect since even if it is presented with the same stimuli several times, it will output specific correction velocities each time. It should be noted that for the situations where controller E does manage the insertion, it does so very elegantly without much wiggling back and forth. This can be seen in Fig. 5.9 which plots the contact force over time during two successful insertions, one with controller D and one with controller E.

5.6.2 GENERALIZATION

To see if the model learned on the steel peg and hole could generalize to insertion with other properties, we evaluated the performance of the best performing method, D, on the wooden peg and rubber hole, see Fig. 5.7d. The friction properties between these materials are significantly different from the steel peg and hole, as was tested manually. Two sets of 20 trials were performed. In the first set of trials, the model trained on data from the steel peg and hole was used. This was contrasted to the performance of a model that was learned with training data from the wooden peg and rubber hole. The results from these experiments are shown in Table 5.2. While the model learned from the steel peg and hole was not as good as the one learned from data on the wooden peg and

Table 5.2: Results from generalization experiment. 20 trials were carried out using adaptation scheme D on the peg and hole in Fig. 5.7d. In this table, D1 represents the model learned from the steel peg and hole, and D2 represents the model learned on the wooden peg and rubber covered hole. Statistics are given as: mean(standard deviation).

	Rate of success, %	Completion time, s	Norm of contact force, N	Norm of contact torque, Nm
D1	55	4.65	4.79 (3.03)	0.46 (0.28)
D2	85	4.42	5.21 (4.56)	0.59 (0.31)

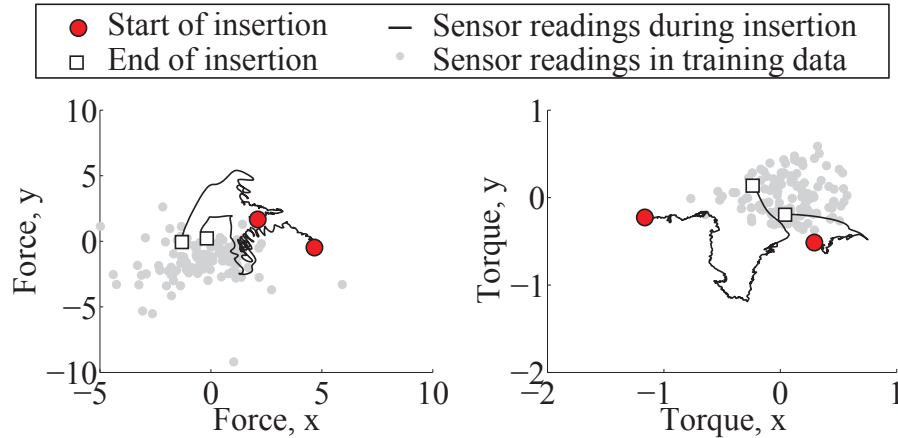


Figure 5.10: The figures show the sensor states visited in training and the sensor traces from two successful executions on the wooden peg with rubber hole. As can be seen, parts of the sensor traces lie quite far from the sensor inputs visited in the training data. **Left:** The sensed force in x and y directions. **Right:** The sensed torque around x and y axes.

hole, the success rate of 55% still indicates some degree of generalization. Fig. 5.10 shows the sensor trace from two successful insertions on the wooden peg and rubber hole using the model learned on the steel peg and hole. Clearly, the sensory inputs experienced during insertion on the second setup lie at least partially outside of the range that was covered during training.

5.7 Discussion and conclusion

In this chapter, we have investigated transfer of task-specific reactive behavior for insertion tasks from humans to robots. We modeled the joint distribution of sensed wrench and corrective angular velocity, and use the distribution over angular velocities conditioned on sensed wrench for autonomously executing the insertion after learning. We compared regression and sampling from this conditional with random corrections not taking the sensed wrench into account. The results indicate a strong advantage to using the learned model, and in particular

to use sampling from the conditional probability rather than regression. This can be seen as a form of guided random search, where not only the mean of the distribution but also its covariance is influenced by the current sensed wrench and depend on the variability of the demonstrations.

Our results show that the proposed approach can generalize to some degree to other insertion tasks. In our view, the ability to generalize to new situations is a crucial aspect of any learning system. In general, one can not expect the learned model to perform as well on a new situation as for that which it was trained. One can, however leverage from a model learned on a different situation by using it as a baseline for a new situation, and adapting it as necessary.

Peg-in-hole is a benchmark problem in robotics and has been tackled by numerous previous works. Comparisons are difficult, because different assumptions are made in different works, different clearance ranges are considered etc. A very early approach that proposed to use recordings of wrench and corrective motion was proposed in (Asada, 1993), which proposed the use of neural networks to model the non-linear map between contact wrench and corrective motion command. However, it is never detailed how the data would be acquired and importantly the proposed method does not allow to use sampling in an informed way — an aspect that in our case significantly increased the performance.

More recent related work exploit passive compliance in combination with hand-coded search patterns for similar problem setup as ours (Balletti et al., 2012; Park et al., 2013). While these works do not need to use a force sensor, however the predetermined search strategy may lead to very high contact forces depending on how the lower level compliance is tuned (Balletti et al., 2012). Also, none of these works show that the hand-coded search strategies can generalize to new insertion tasks. If the programmed search pattern must be adjusted to each task, it is a time consuming process which is certainly counter-intuitive to non-expert robot operators. Our approach is in this sense more user friendly since all that is required for performing a new insertion tasks is that a few demonstrations are provided. Also, monitoring of contact wrenches allows to protect the parts in fragile assembly operations. While our method does not guarantee low contact wrench, our contact wrenches are lower than those reported in (Balletti et al., 2012) and a safety mechanism to abort the insertion and start a new attempt in the event of an excessive contact wrench could easily be added to our system if necessary.

In (Abu-Dakka et al., 2014), a system for learning peg-in-hole in terms of desired pose and wrench profiles is proposed. Trajectories are modeled with DMP, and a coupling terms are used 1) to slow down the system in the absence of significant deviations from the expected value and the desired value 2) to adapt the position so as to achieve the desired contact force. The latter is also done iteratively using an ILC scheme. In our experiments, we have found that the jamming condition that invariably appears with small enough clearance results

in contact forces that are generally not repeatable. Hence, approaches such as that offered in (Abu-Dakka et al., 2014) are useful for generalizing the initial part assembly to new configurations etc, but it is unclear if they can resolve the type of jamming problem that we addressed in this chapter. An interesting approach would be a combination where the system like that in (Abu-Dakka et al., 2014) replaces the hand-coded approach phase (Section 5.5.1) while our system can be used in the final insertion phase.

In this work, the human was used as a teacher for showing how the robot should react to readings on its force sensor. This allowed us to learn a controller that performs reasonably well, but it is still far behind the performance of the human on the same insertion setup. Ongoing research in our group¹ studies human search and insertion behaviors when deprived of auditory and visual sensory input. We expect that this study will shed light on what it is that allows the human to outperform robots so strongly in these tasks, as well as outline how the present control scheme can be further improved, perhaps with varying stiffness or search strategies that are more generic than the task-specific ones we presented here. Furthermore, considering corrections of translational motion as well as orientation corrections could increase the flexibility of the proposed approach.

This chapter was focused on a specific type of compliant manipulation tasks, which require active adaptation of the motion in response to the contact wrench. It differed from the architecture in Chapters 3 and 4 in the reference trajectory is updated online in response to sensory input. In general, the idea of online modification of the reference trajectory generation or modification is a necessity for many manipulation tasks in unstructured environments. The next chapter continues in the same spirit of online responsiveness of the reference trajectory but instead of adaptation to externally sensed information the internal state of the robot is used to generate the trajectory.

¹Learning Algorithms and Systems Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland

INCREMENTAL MOTION LEARNING WITH LOCALLY MODULATED DYNAMICAL SYSTEMS

6.1 Introduction

In Chapters 3 and 4, the availability of a trajectory for the task was assumed. The trajectory was generally represented by time-indexed trajectories. In chapter 5 the reference trajectory was generated online in response to experienced contact forces. In this chapter, we explore online motion generation but instead generating a desired velocity as a function of the current position. Such motion representation was already used in Section 3.3.3, using the SEDS algorithm for batch learning of a DS representation via GMR. In this chapter we turn to the problem of incrementally learning a DS representation of a task using LfD.

In order to successfully model a robot motion, demonstrations should be provided such that they include the generic characteristics of the motion. This is however very difficult when dealing with complex and/or high dimensional motions. Incremental Learning, whereby the robot successively learns the task through several demonstrations, can alleviate this difficulty. Furthermore, incremental learning can allow task refinement (incremental adaptation of task model to improve task performance) and reuse (use of an existing task model for a completely different task) (Sauser et al., 2011). A general work-flow of an incremental learning setting is described in Fig. 6.1. While numerous advances have been made for incremental motion learning for time-indexed trajectories, incremental learning in DS representations is still a largely unexplored area of research.

In this work, we address this by proposing a novel DS representation, called Locally Modulated Dynamical Systems (LMDS), that allows to reshape a DS while preserving stability properties of the original system. As hinted by the name, this is done by locally applying transformations (e.g. rotations), to the original dynamics. It is shown that this way of reshaping dynamics is suitable for robot motion modeling, since complex motions can be modeled without risking the introduction of spurious attractor points or unstable behavior. The LMDS representation is not constrained to a particular form of original dynamics or learning method (any *local* regression method can in principle be used, together with any representation of a first order *autonomous* dynamical system). We further propose the Gaussian Process Modulated Dynamical Systems (GP-MDS)

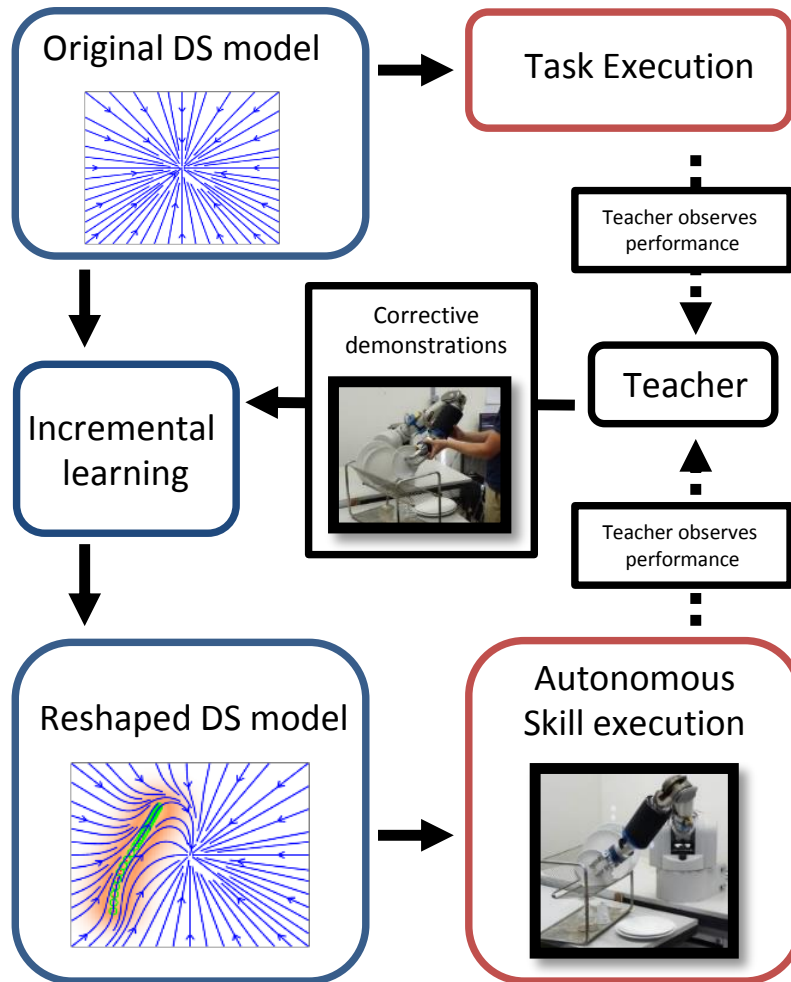


Figure 6.1: The figure illustrates the incremental process of acquiring a new skill by reshaping dynamics using the proposed framework. The system can be reshaped repeatedly until satisfactory task performance has been achieved.

algorithm, which uses Gaussian Process Regression (GPR) to learn reshaped dynamics. The contents of this chapter have been published in (Kronander et al., 2015).

Related work to this chapter is reviewed in Section 2.1.7. The remainder of this chapter is organized as follows. In Section 6.2 we detail the LMDS formalism, and propose a particular parameterized form of the modulation function which is used in this work. In Section 6.3, we then address the problem of how to learn LMDS, introducing the GP-MDS algorithm. Experimental validation is presented in Section 6.4, with a 2d example of warping dynamics for handwriting letters, and one periodic as well as one discrete manipulation task on the KUKA LWR and Barret WAM arms. The chapter is concluded with a discussion in Section 6.5.

6.2 Locally Modulated Dynamical Systems

In this work, we assume the availability of an autonomous DS which serves as a coarse map of desired trajectories of a robotic task. We will refer to this DS as the *original dynamics*. In this work, we will exclusively use original dynamics that are asymptotically stable at a single attractor point.

6.2.1 FORMULATION AND PROPERTIES

Let $\boldsymbol{\xi} \in \mathbb{R}^N$ represent a N -dimensional kinematic variable, e.g. a Cartesian position vector. Let a continuous function $\mathbf{f} : \mathbb{R}^N \mapsto \mathbb{R}^N$ represent the original dynamics:

$$\dot{\boldsymbol{\xi}} = \mathbf{f}(\boldsymbol{\xi}) \tag{6.1}$$

These dynamics are reshaped by a continuous matrix-valued function $\mathbf{M}(\boldsymbol{\xi})$ that modulates the original dynamics:

$$\dot{\boldsymbol{\xi}} = \mathbf{g}(\boldsymbol{\xi}) = \mathbf{M}(\boldsymbol{\xi})\mathbf{f}(\boldsymbol{\xi}) \tag{6.2}$$

As will be shown later, this seemingly simple representation is highly flexible and can model very complex motions. We will only consider modulation functions that vary smoothly across the state-space, i.e. no abrupt changes that could affect the resulting system in a counter-intuitive manner are allowed. If the modulation is local and full rank, several important properties are inherited from the original to the reshaped dynamical system. Before listing some key properties of the LMDS formulation, we introduce the concept of a *locally active* matrix function.

Definition 6.1 (Locally active). A matrix-valued function $\mathbf{M}(\boldsymbol{\xi}) \in \mathbb{R}^{N \times N}$ is said to be *acting locally* or to be *locally active* if there exists a compact subset $\chi \subset \mathbb{R}^N$ such that $\mathbf{M}(\boldsymbol{\xi}) = \mathbf{I}_{N \times N}$ for all $\boldsymbol{\xi} \in \mathbb{R}^N \setminus \chi$.

Proposition 6.1 (Equilibrium points). *If $\mathbf{M}(\boldsymbol{\xi})$ has full rank for all $\boldsymbol{\xi}$, the reshaped dynamics, Eq. (6.2) has the same equilibrium point(s) as the original dynamics, Eq. (6.1).*

If $\mathbf{M}(\boldsymbol{\xi})$ has full rank, it has an empty null-space, and hence Eq. (6.2) is zero iff $\mathbf{f}(\boldsymbol{\xi}) = \mathbf{0}$. This simple result is of tremendous importance for using DS for motion representation. The introduction of spurious attractors is one of the main problems in using regression methods to learn dynamics (Khansari-Zadeh and Billard, 2011). Here, we make such spurious attractors impossible by construction.

Proposition 6.2 (Boundedness). *Assume that the original dynamics is bounded (see Def. 2.1). Assume further that $\mathbf{M}(\boldsymbol{\xi})$ is locally active in a compact subset $\chi \subset \mathbb{R}^N$ (see Def. 6.1). Then, the reshaped dynamics is bounded.*

Proof. Let B_R be a ball centered at the origin of radius R in \mathbb{R}^N . Let R be chosen such that χ lies entirely in B_R . Since χ is a compact set in \mathbb{R}^N , it is always possible to find such a R . For each $\delta > 0$, let $\epsilon(\delta) > 0$ be an associated boundary for the original dynamics (refer to Def. 2.1). Define $\epsilon'(\delta)$ as a boundary for the reshaped dynamics as follows: $\epsilon' = \epsilon(R)$ for $\delta < R$ and $\epsilon' = \epsilon(\delta)$ for $\delta \geq R$. Boundedness follows from Def. 2.1. \square

DS that have a single attractor are useful for representing generic motions to a fixed point, e.g. reach-and-grasp type motions. For such systems, in addition to equilibrium points and boundedness, the stronger stability property is also inherited, and locally the stability is asymptotic.

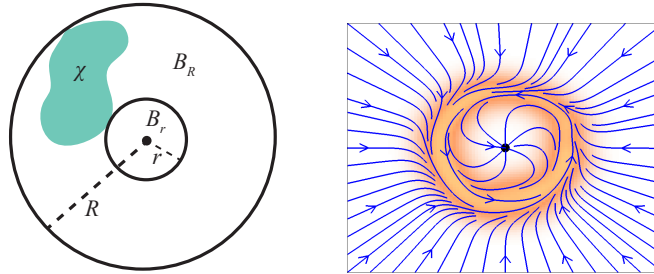


Figure 6.2: **Left:** Illustration of the introduction of the balls B_r and B_R . B_r is the ball centered at the origin with the largest possible r so that there are no points of χ in B_r . B_R is a ball with radius R , which is chosen as the smallest possible number so that B_R fully contains χ . **Right:** An example of a 2D reshaped system where the original asymptotically stable (linear) dynamics are reshaped to a system that converges to either a limit cycle or the origin depending on starting position. The reshaped system is globally stable but not *asymptotically* stable.

Proposition 6.3 (Lyapunov stability). *Consider a system $\dot{\boldsymbol{\xi}} = \mathbf{f}(\boldsymbol{\xi})$ that has a single equilibrium point. Without loss of generality, let this equilibrium point be placed at the origin. Assume further that the equilibrium point is stable. Assume*

that the criteria for Propositions 1 and 2 are satisfied. If in addition, χ does not include the origin, the reshaped system is stable at the origin.

Proof. According to Proposition 1, the reshaped dynamics has a single equilibrium point at the origin. Let B_r be a ball centered at the origin with a radius r small enough that B_r does not include any point in χ . Hence, inside B_r , we have $\mathbf{g}(\boldsymbol{\xi}) = \mathbf{f}(\boldsymbol{\xi})$. By the stability of \mathbf{f} , there exists for all $0 < \epsilon < r$ a $\delta(\epsilon)$ such that $\|\boldsymbol{\xi}(0)\| < \delta(\epsilon) \Rightarrow \|\boldsymbol{\xi}(t)\| < \epsilon, \forall t > 0$. For any $\epsilon > r$, let $\delta(\epsilon) = \delta(r)$. Then, by the stability of \mathbf{f} , $\|\boldsymbol{\xi}(0)\| < \delta(\epsilon) = \delta(r) \Rightarrow \|\boldsymbol{\xi}(t)\| < r < \epsilon$. □

Note that the above proposition does not ensure *asymptotic* stability, which would mean that all trajectories are guaranteed to converge to the origin. Instead, Prop. 6.3 says that trajectories can be kept arbitrarily close to the origin, if they start close enough. Since stability is necessary for asymptotic stability, the result is important because it represents a minimum requirement, but not a guarantee, for trajectories to converge to attractor of the original system. Unsurprisingly, the precondition that there is a region around the origin which is not reshaped also implies local asymptotic stability.

Proposition 6.4 (Local asymptotic stability). *Consider a system $\dot{\boldsymbol{\xi}} = \mathbf{f}(\boldsymbol{\xi})$ that has a single equilibrium point. Assume that the conditions of Propositions 1, 2 and 3 are satisfied. Then, the reshaped system is locally asymptotically stable at the origin.*

Proof. The original dynamics are globally asymptotically stable, which implies the existence of a Lyapunov function $V : \mathbb{R}^N \mapsto \mathbb{R}^+$ such that:

$$V(\boldsymbol{\xi}) > 0, \forall \boldsymbol{\xi} \neq 0 \quad \text{and} \quad V(0) = 0 \quad (6.3)$$

$$\dot{V} = \frac{\partial V}{\partial \boldsymbol{\xi}} \mathbf{f}(\boldsymbol{\xi}) < 0, \forall \boldsymbol{\xi} \neq 0 \quad \text{and} \quad \dot{V}(0) = 0 \quad (6.4)$$

Let B_r be defined as in the proof of Proposition 6.3. Let $\mathcal{S} \subset B_r$ denote the largest level set of V that lies entirely inside B_r . For any $\boldsymbol{\xi}_0 \in \mathcal{S}$, the reshaped dynamics is exactly equal to the original dynamics $\dot{\boldsymbol{\xi}} = \mathbf{f}(\boldsymbol{\xi})$. Hence, $V(\boldsymbol{\xi}) > 0$ and $\dot{V}(\boldsymbol{\xi}) < 0$ holds for all $\boldsymbol{\xi} \in \mathcal{S}$, which proves that the system is locally asymptotically stable at the origin with region of attraction given by \mathcal{S} . □

If demonstrations are given that clearly contradict the asymptotic stability property of the original dynamics, it will not be retained. A simple example is given in Fig. 6.2, which illustrates a reshaped asymptotically stable linear system. As can be seen, the resulting topology after reshaping is a half-stable¹ limit cycle. The non-inheritance of global asymptotic stability is both an advantage and a disadvantage. It is an advantage because it allows to represent repetitive

¹ The term half-stable refers to the property that trajectories may converge to the limit cycle or an attractor point depending on the starting location.

motions such as shaking or polishing, as will be shown in Section 6.4.3. It is a disadvantage because for discrete motions, it would be preferable to rigorously ensure that the target point will be reached under all circumstances. However, we conjecture that limit cycles are not introduced unless they are explicitly demonstrated, as will be exemplified in Section 6.4.2.

6.2.2 ILLUSTRATIVE EXAMPLES

Here, we give set of illustrative 2d examples of some types of transformations that can be achieved in the LMDS formulation. Consider the following linear original dynamics:

$$\dot{\boldsymbol{\xi}} = - \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \boldsymbol{\xi} \quad (6.5)$$

The following function will be used to control the influence of the modulations, i.e. in what region of the state-space they influence the dynamics.

$$h(\boldsymbol{\xi}) = \exp(-50\|\boldsymbol{\xi} - \mathbf{c}\|^2) \quad (6.6)$$

Let the center point of the influence function be placed at $\mathbf{c} = [50, 50]^T$. This function is used to impose the locally active property², see Def. 6.1.

LOCAL MODULATION BY RANDOM MATRIX

For illustrative purposes, we construct here a LMDS which will locally modulate the original dynamics with a random matrix, $A \in \mathbb{R}^{2 \times 2}$. We define the modulation as follows:

$$\mathbf{M}_a(\boldsymbol{\xi}) = (1 - h(\boldsymbol{\xi}))I_2 + h(\boldsymbol{\xi})A \quad (6.7)$$

This modulation is locally active³, but does not have full rank everywhere. Consequently, the modulation can introduce spurious attractors, as is illustrated in Figures 6.3a- 6.3c.

LOCALLY ROTATING DYNAMICS

One particularly interesting class of local modulations are rotations of the original dynamics. Let $\phi(\boldsymbol{\xi}) = h(\boldsymbol{\xi})\phi_c$ denote a state-dependent rotation angle. This results in a smoothly decaying rotation which will fully rotate the dynamics by the angle ϕ_c only at $\boldsymbol{\xi} = \mathbf{c}$. The modulation function is then defined as the associated rotation matrix:

$$\mathbf{M}_r(\boldsymbol{\xi}) = \begin{bmatrix} \cos(\phi(\boldsymbol{\xi})) & -\sin(\phi(\boldsymbol{\xi})) \\ \sin(\phi(\boldsymbol{\xi})) & \cos(\phi(\boldsymbol{\xi})) \end{bmatrix} \quad (6.8)$$

²Strictly speaking, for the modulation to be locally active, Eq. (6.6) should be set to zero at a small value, as described in Section 6.3.3. This is, however, for illustrative purposes not necessary here.

³See previous footnote.

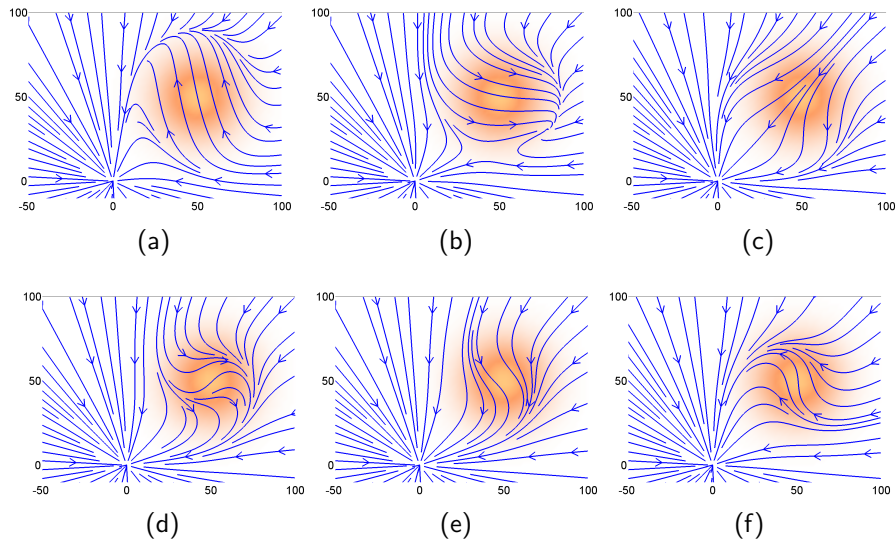


Figure 6.3: **Top:** Examples of local modulation of linear dynamics with three different random matrices. Note that in the second example (top-middle), a spurious attractor point has been introduced due to rank-deficiency of the modulating function. **Bottom:** Examples of local rotation of linear dynamics with three different rotation angles.

In this case, $\mathbf{M}_r(\boldsymbol{\xi})$ is guaranteed to have full rank for all $\boldsymbol{\xi}$. Furthermore, the modulation is locally active⁴. This means that local rotations can be applied without introducing spurious attractors, regardless of the form of the original dynamics. This very useful property will be exploited to apply non-parametric learning without constraints, as is detailed in Section 6.3. Examples of locally rotating the linear dynamics in Eq. (6.5) with a few different values of ϕ_c are given in Figures 6.3d- 6.3f

6.2.3 MODULATION BY ROTATION AND NORM-SCALING

In this section, we describe a particular choice of modulation function which is used in the remainder of this chapter. As seen in Section 6.2.2, rotations (or any other orthogonal transformations) always have full rank. It is possible to define and parameterize rotations in any dimension, but we will focus mainly on 2d and 3d systems in the remainder of this work.

For increased flexibility, a scaling of the speed in the DS can be achieved by multiplying the rotation matrix by a scalar. Let $\mathbf{R}(\boldsymbol{\xi})$ denote a state-dependent rotation matrix, and let $\kappa(\boldsymbol{\xi})$ denote a state-dependent scalar function strictly larger than -1 . We then construct a modulation function that can locally rotate and speed up or slow down dynamics as follows:

$$\mathbf{M}(\boldsymbol{\xi}) = (1 + \kappa(\boldsymbol{\xi}))\mathbf{R}(\boldsymbol{\xi}) \quad (6.9)$$

⁴Strictly speaking, for the modulation to be locally active, Eq. (6.6) should be set to zero at a small value, as described in Section 6.3.3.

Table 6.1: Procedure for converting 2d or 3d trajectory data to modulation data.

Input: Trajectory data $\{\boldsymbol{\xi}_m, \dot{\boldsymbol{\xi}}_m\}_{m=1}^M$

- 1: **for** $m = 1$ **to** M **do**
- 2: Compute original velocity: $\dot{\boldsymbol{\xi}}_m^o = \mathbf{f}(\boldsymbol{\xi}_m)$
- 3: Compute rotation vector (3d only): $\boldsymbol{\mu}_m = \frac{\dot{\boldsymbol{\xi}}_m \times \dot{\boldsymbol{\xi}}_m^o}{\|\dot{\boldsymbol{\xi}}_m\| \|\dot{\boldsymbol{\xi}}_m^o\|}$
- 4: Compute rotation angle: $\phi_m = \arccos \frac{\dot{\boldsymbol{\xi}}_m^T \dot{\boldsymbol{\xi}}_m^o}{\|\dot{\boldsymbol{\xi}}_m\| \|\dot{\boldsymbol{\xi}}_m^o\|}$
- 5: Compute scaling $\kappa_m = \frac{\|\dot{\boldsymbol{\xi}}_m\|}{\|\dot{\boldsymbol{\xi}}_m^o\|} - 1$
- 6: 3d: $\boldsymbol{\theta}_m = [\phi_m \boldsymbol{\mu}_m, \kappa_m]$ 2d: $\boldsymbol{\theta}_m = [\phi_m, \kappa_m]$
- 7: **end for**
- 8: **return** Modulation data $\{\boldsymbol{\xi}_m, \boldsymbol{\theta}_m\}_{m=1}^M$

Both κ and \mathbf{R} should vary in a continuous manner across the state-space. In a continuous system, the inclusion of a speed-scaling does not influence the stability properties, although it may do so in discrete implementations, so care should be used to not allow $\kappa(\boldsymbol{\xi})$ to take large values. Also, note that κ has been given an offset of 1 so that with $\kappa(\boldsymbol{\xi}) = 0$ the original speed is retained. This is useful when modeling κ with local regression techniques such as GPR, as will be done in Section 6.3.2.

Rotations in arbitrary dimension can be defined by means of a two-dimensional *rotation set* and a rotation angle ϕ . In 2d, the fact that the rotation set is the entire \mathbb{R}^2 means that a rotation is fully defined by the rotation angle only. Hence, the parameterization in that case is simply $\boldsymbol{\theta}_{2d} = [\phi, \kappa]$. In 3d, the rotation plane can be compactly parameterized by its normal vector. Hence, the parameterization in 3d is $\boldsymbol{\theta}_{3d} = [\phi \boldsymbol{\mu}_R, \kappa]$, where $\boldsymbol{\mu}_R$ is the rotation vector (the normal of the rotation set). Parameterizations in higher dimensions are possible, but require additional parameters for describing the rotation set.

6.3 Learning Locally Modulated Dynamical Systems

In the previous section, we described how the dynamics can be reshaped in the LMDS framework. We now turn to the problem of how to learn from data in LMDS. The procedure for generating training data for LMDS from trajectory data is described in Section 6.3.1. After this step, one can in principle use any local regression technique to learn an LMDS system.

6.3.1 TRAINING DATA

Assume that a training set of M observations of $\boldsymbol{\xi}$ and $\dot{\boldsymbol{\xi}}$ is available: $\{\boldsymbol{\xi}_m, \dot{\boldsymbol{\xi}}_m\}_{m=1}^M$. To exploit this data for learning, it is first converted to a data set consisting of input locations and corresponding modulation vectors: $\{\boldsymbol{\xi}_m, \boldsymbol{\theta}_m\}_{m=1}^M$. To compute the modulation data, the first step is to compute the *original velocities*,

denoted by $\dot{\xi}_m^o, m = 1 \dots M$. These are computed by evaluating the original dynamics function at all $\xi_m, m = 1 \dots M$ in the trajectory data set. Each pair $\{\dot{\xi}_m^o, \xi_m\}$ then corresponds to a modulation parameter vector θ_m . How this parameter vector is computed depends on the structure and parameterization chosen for the modulation function. The procedure for computing the modulation parameters for the particular choice of modulation function used in this work (rotation and norm scaling) is described in Table 6.1. Parameter vectors for each collected data point are computed this way and in pairs with the corresponding state-observations constitute a new data set: $\{\xi_m, \theta_m\}_{m=1}^M$. Regression can now be applied to learn $\theta(\xi)$ as a state-dependent function.

6.3.2 GAUSSIAN PROCESS MODULATED DYNAMICAL SYSTEMS

Gaussian Process Regression (GPR), is a state-of-the-art regression technique which in its standard form can model functions with input of arbitrary dimension and scalar outputs. Some background on GPR is provided in Section 2.2.5.

The behavior of GPR depends on the choice of covariance function $k(\cdot, \cdot)$. In this work, we use the squared exponential covariance function, defined by:

$$k(\xi, \xi') = \sigma_f \exp\left(-\frac{\|\xi - \xi'\|^2}{2l}\right)$$

where $l, \sigma_f > 0$ are scalar hyper-parameters. In this work, these parameters are set to predetermined values. Alternatively, they could be optimized to maximize the likelihood of the training data (Rasmussen and Williams, 2006).

GP-MDS is based on encoding the parameter vector of the modulation function with Gaussian Processes. The data set from Section 6.3.1 is used as training set for the GP, where the positions ξ_m are considered as inputs and the corresponding modulation parameters θ_m are considered as outputs. Note that since θ is multidimensional, one GP per parameter is needed. This can be done at little computational cost if the same hyper-parameters are used in each GP, as is clear by inspecting Eq. (2.33). A vector of scalar weights can be pre-computed:

$$\alpha(\xi^*) = (\mathbf{k}^*[\mathbf{K} + \sigma_n^2 I]^{-1})^T \quad (6.10)$$

Prediction of each entry of θ then only requires computing a dot-product: $\hat{\theta}^j(\xi^*) = \alpha(\xi^*)^T \Theta^j$, where Θ^j is a vector stacking all the training samples of the j th parameter of θ .

6.3.3 ENFORCING LOCAL MODULATION

The particular choice of GP prior with zero mean and with the squared exponential covariance function results all elements of θ going to zero in regions far from any training data. Hence, for the modulation to be local, it should be parameterized such that $\mathbf{M} \rightarrow I$ as $\theta \rightarrow 0$. This is the case for the rotated

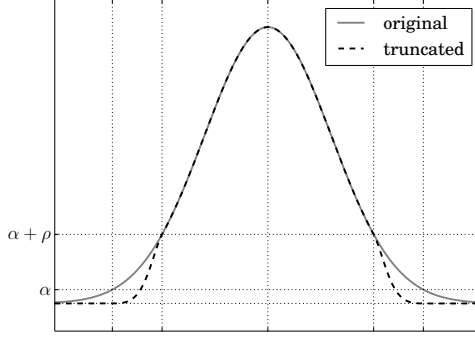


Figure 6.4: The figure illustrates the smooth truncation function in Eq. (6.11) on an Gaussian kernel. For clarity of illustration, the threshold has been set to a relatively high value.

and speed-scaled modulation that are used in this work, which as described in Section 6.3.1 encode the rotation angle in the norm of a sub-vector of $\boldsymbol{\theta}$. Also, when the speed factor κ goes to zero, the speed of the reshaped dynamics approaches the original speed. Consequently, the modulation function does go to identity, but there is no strict boundary outside of which \mathbf{M} is exactly equal to \mathbf{I} . To make the modulation locally active in the strict sense, the entries of $a(\boldsymbol{\xi}^*)$ (Eq. (6.10)) should be smoothly truncated at some small value. To this end, we used a sinusoid computing the truncated weights $\alpha'(\boldsymbol{\xi}^*)$ as follows:

$$\alpha'(\boldsymbol{\xi}^*) = \begin{cases} 0 & \alpha(\boldsymbol{\xi}^*) < \underline{\alpha} \\ \frac{1}{2} \left(1 + \sin \left(\frac{2\pi(\alpha(\boldsymbol{\xi}^*) - \underline{\alpha})}{2\rho} - \frac{\pi}{2} \right) \right) \alpha(\boldsymbol{\xi}^*) & \underline{\alpha} \leq \alpha(\boldsymbol{\xi}^*) \leq \underline{\alpha} + \rho \\ \alpha(\boldsymbol{\xi}^*) & \underline{\alpha} + \rho < \alpha(\boldsymbol{\xi}^*) \end{cases} \quad (6.11)$$

This function is illustrated in Fig. 6.4. Throughout this work, we used the weighting function above with values $\underline{\alpha} = 0.01$ and $\rho = 0.01$. It should be noted that this particular choice of truncation function is not critical, and could surely be replaced by other methods without any perceivable impact on the resulting dynamics. The computation of the reshaping parameters $\hat{\theta}^j(\boldsymbol{\xi}_*)$ at a query location $\boldsymbol{\xi}^*$ can hence be summarized as follows:

1. compute $\boldsymbol{\alpha}(\boldsymbol{\xi}^*)$ according to Eq. (6.10)
2. compute the truncated weights according to Eq. (6.11)
3. compute the predicted parameters $\hat{\theta}^j(\boldsymbol{\xi}_*) = \boldsymbol{\alpha}'(\boldsymbol{\xi}_*)^T \boldsymbol{\Theta}^j$

6.3.4 TRAJECTORY-BASED SPARSITY CRITERIA

If fixed hyper parameters are considered as in this work, incremental learning can be achieved simply by expanding the training set used for GPR. To deal with the increased complexity of having to recompute a $M \times M$ matrix inverse each time a new data point is added, it is useful to sparsely represent the incoming data. This is often referred to as selecting an *active set* of training points. The GP literature is rich in sparse approximations. As complete coverage of all proposed sparse GP methods lies outside the scope of this work, we refer the reader to (Quiñonero Candela et al., 2005) which provides an excellent review and unifying view of most sparse approximations. What most previous methods have in common is that they define sparsity criteria that are based on the input patterns of the data. This is natural in the GP framework, since the inputs implicitly define the covariance of the outputs and hence allow to use information-theoretic principles for selecting the data to use for regression. In contrast, we define a sparsity criterion based on the outputs, similarly to e.g. the LWPR algorithm (Vijayakumar et al., 2005). This criterion is designed to select training points not to maximize information gain, but to maximize performance metrics that are important for the specific application of trajectory encoding with DS.

Assume that there is already M training data points in the GP training set. In order to determine if a new data point $\{\boldsymbol{\xi}_{M+1}, \boldsymbol{\theta}_{M+1}\}$ should be included in the GP training set, we introduce two functions:

$$J_{M+1}^1 = \frac{|\kappa_{M+1} - \hat{\kappa}(\boldsymbol{\xi}_{M+1})|}{1 + \kappa_{M+1}} \quad (6.12a)$$

$$J_{L+1}^2 = \min_{k \in \mathbb{N}} (|\phi_{M+1} - \hat{\phi}(\boldsymbol{\xi}_{M+1}) + 2k\pi|) \quad (6.12b)$$

where $\hat{\kappa}(\boldsymbol{\xi}_{M+1})$ and $\hat{\phi}(\boldsymbol{\xi}_{M+1})$ denote the predicted speed scaling and rotation angle at the new input points, using the *existing* GP training data $\{\boldsymbol{\xi}_m, \theta_m\}_{m=1}^M$. Eq. (6.12a) is a relative measure of the speed error, and (6.12b) is an absolute measure of the error in rotation angle. The data point is added to the training set if either of J_{M+1}^1 or J_{M+1}^2 exceed predetermined threshold values \bar{J}^1, \bar{J}^2 . E.g. by setting $\bar{J}^1 = 0.1$ and $\bar{J}^2 = 0.1\pi$, speed errors of less than 10% and error in rotation angle below 0.1π are considered acceptable. Note that these thresholds relate directly to the trajectory and are hence easily tuned to a desirable trade-off between sparsity and accurate trajectory representation.

An illustrative example of GP-MDS on toy 3d data is given in Fig. 6.5.

6.4 Evaluation

In this section, we present simulations and experiments to evaluate the proposed approach. First, GP-MDS is applied for refining handwriting motions in 2d. We then provide a set of simulations with artificially generated data illus-

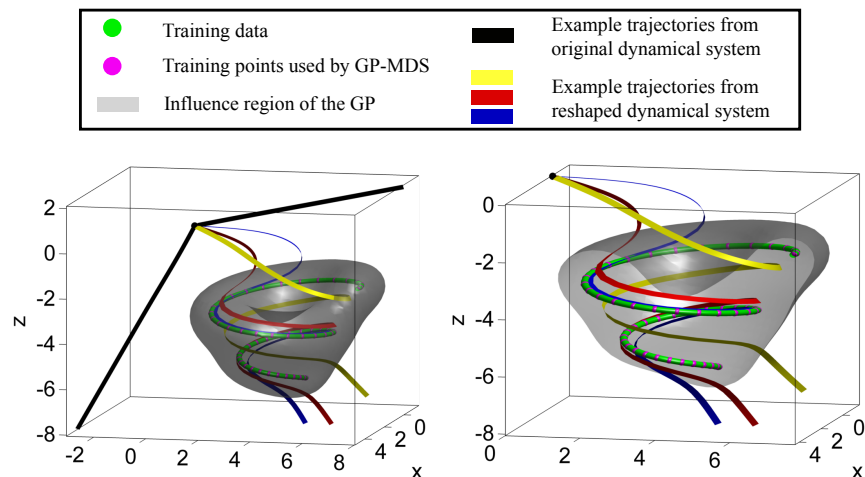


Figure 6.5: Left: Example of reshaped dynamics using GP-MDS in a 3d system. The colored stream-tapes represent example trajectories of the reshaped dynamics. The stream-tapes colored in black represent trajectories that do not pass through the reshaped region of the state space, and hence retain the straight-line characteristics of the linear system that is used as original dynamics here. The green stream-tube are artificially generated data representing an expanding spiral. Points in magenta represent the subset of this data that was selected as training set. The gray surface illustrates the region in which the dynamics are significantly altered (corresponding to a level set of the predictive variance in the GP). The colored stream-tapes are example trajectories that pass through the reshaped region. **Right:** Same as left but zoomed in and the influence surface has been sliced to improve visibility of the training points and the trajectories.

trate 1) that GP-MDS can represent cyclic motions and 2) cyclic behavior needs to be explicitly demonstrated in order to occur. Lastly, we apply GP-MDS on a real-world task consisting in teaching a robot to put plates in the slots of a dishwasher rack.

6.4.1 HANDWRITING MOTIONS

The LASA handwriting data set (Khansari-Zadeh and Billard, 2011) is commonly used for benchmarking for learning in autonomous DS (Neumann et al., 2013; Lemme et al., 2013). It consists of a series of demonstrated 2d trajectories of handwriting letters. Here, we will not present a comparative evaluation of GP-MDS versus other methods for learning these motions, but focus instead on illustrating an interesting application scenario of GP-MDS for reshaping existing DS with additional demonstrations.

The first column of Fig. 6.6 shows training data for four letters from the LASA handwriting set, along with streamlines from SEDS models trained on this data. Note that these models already do a good job at modeling the data. GP-MDS was applied to refine these dynamics using additional demonstrations. The middle column of Fig. 6.6 shows GP-MDS being applied for modifying the SEDS dynamics. In the case of the S-Shape, starting the trajectory from some points result in a letter with disproportionate features as illustrated by the black trajectory in Fig. 6.6a. In Fig. 6.6b, the dynamics have been reshaped such that trajectories starting in one problematic region are deviated toward a region of the state-space from which they produce a good letter. This results in a trajectory that produces a good letter S after it has been deviated toward the region were the original demonstrations start. For letter N, starting trajectories left of the demonstrated starting location is problematic, as illustrated by the black example trajectory in Fig. 6.6d. In Fig. 6.6e, this is again remedied with a very simple corrective demonstration. For letters W and Z, one additional demonstration (different from the demonstrations used for the SEDS model) was given. The goal here is to sharpen the corners, which are overly smooth both in the original demonstrations and the resulting SEDS model (Figures 6.6g and 6.6j). In order to favor detail over generalization, a fine lengthscale was selected, resulting in the sharpened letters in Figures 6.6h and 6.6k.

The right column of Fig. 6.6 shows streamlines from GP-MDS applied to a linear system in place of an SEDS model. In these cases, the original training data (the same that was used for training the SEDS models) was used training GP-MDS. A medium scale lengthscale was chosen to trade-off generalization and detail. As is seen, in most cases GP-MDS reproduce the shape of the letters, although using considerably more parameters than the SEDS models. While we can conclude that relatively complex motions can be learned even without any knowledge of the task in the original dynamics, the performance of GP-MDS

is better if the original dynamics can already provide a rough estimate of the trajectory.

Note the sparse selection of training data in Fig. 6.6, middle column. In areas of the state-space where the original dynamics have the same direction as the corrective demonstration, it is not necessary to add training data⁵. The sparse data selection is also clearly visible near the end of the letters in Figures 6.6b and 6.6i, since the demonstrations there are roughly aligned with the trajectories of the linear system which is used as original dynamics in these cases.

6.4.2 NON-CONVERGENCE IN RESHAPED SYSTEMS

Recall from Section 6.2.1 that starting from an asymptotically stable DS, reshaping the system with full rank and locally active modulation function (e.g. GP-MDS) only guarantees that the system remains stable. Hence, it is theoretically possible that trajectories could end up in orbits (open or closed) instead of converging to the attractor point of the original system. We argue that this is not a problem, because in practice GP-MDS converges unless it is presented with data that explicitly indicates orbital behavior. In this section, we support this statement by providing GP-MDS with data that is artificially generated to be at risk of generating such behavior in the reshaped system.

Fig. 6.7a shows a GP-MDS reshaping a linear system explicitly trying to create a repetitive pattern. The resulting system converges either to a limit cycle or the origin, depending on the starting point of the trajectory. In Fig. 6.7b, additional data has been provided that changes the characteristic of the origin from a sink to a source, resulting in a system in which all trajectories converge to the stable limit cycle. Note that the system in Fig. 6.7b is not stable at the origin, and violates a condition of Prop. 6.3, because the reshaped region includes the origin. In planar systems, any closed orbits necessarily enclose an equilibrium point, which is illustrated in Fig. 6.7c where similar data has been presented in a different part of the plane. The resulting system is asymptotically stable at the origin, although it exhibits characteristics that are undesirable for any practical application. We retain from this that closed orbits in the demonstrations should generally be avoided.

In higher dimension orbits can in principle occur anywhere in the state-space. We have found that it is quite difficult to produce data that causes orbital behavior, although it is possible. Fig. 6.8 shows GP-MDS used to reshape a linear system in 3d with artificial data from a spiral shaped motion. Even with a very tight spiral as in Fig. 6.8a, it seems that trajectories do not get stuck but would eventually converge to the origin. Only when translating the data so that the spiral has the origin in its center, and by significantly increasing the lengthscale of the GP were we able to clearly produce a system in which the

⁵In these experiments, $\overline{\mathcal{J}}^1$ was set to a very high value, since speed was not considered important for this task. Hence, selection criteria is in practice only depending on the directionality of the vector field.

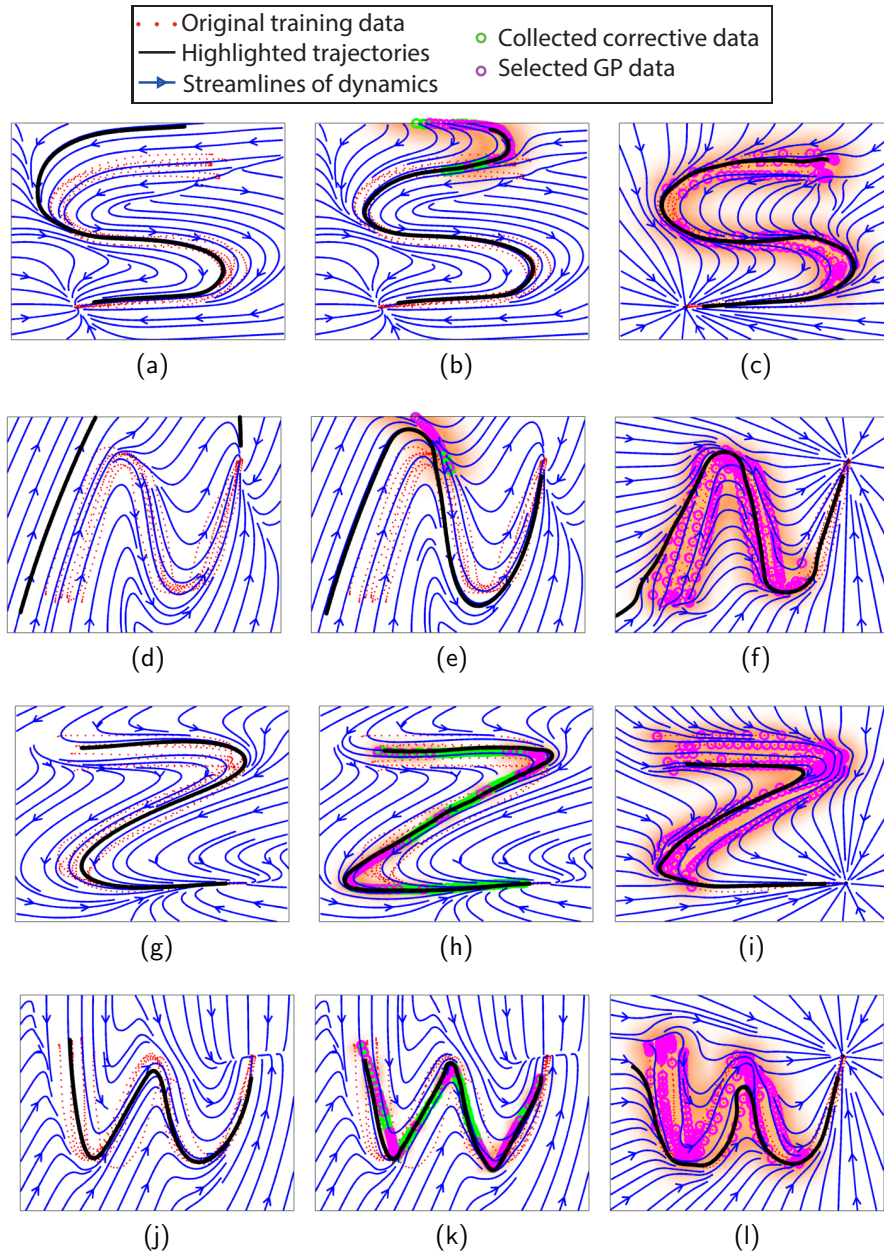


Figure 6.6: **Left column:** Demonstrated trajectories and resulting SEDS models for the letters S,N,Z and W. **Middle column:** GP-MDS is used to improve various aspects of the SEDS models. In the case of letters S and N the favorable starting region of the state space is achieved with very simple data being provided to GP-MDS with a crude length-scale. In the case of Z and W, GP-MDS with a fine length-scale is used to sharpen the corners of the letters. **Right column:** The original training data is provided to GP-MDS, with a simple linear system replacing SEDS as original dynamics.

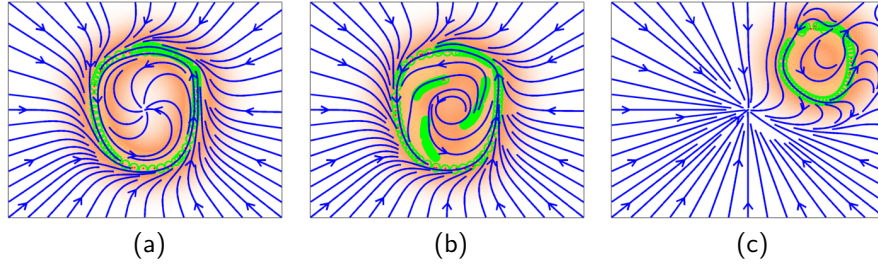


Figure 6.7: **a)** A demonstration of a closed repetitive pattern is used for reshaping a linear 2d system with GP-MDS. **b)** An additional demonstration aimed at destabilizing the origin results in a system in which all trajectories converge to a stable limit cycle. **c)** A demonstrated closed curve not containing an equilibrium point.

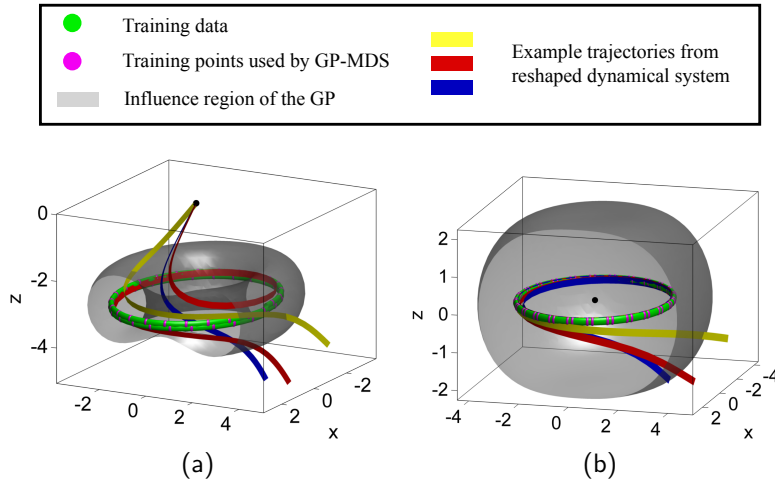


Figure 6.8: Left: An artificially generated spiral-shaped trajectory is used to reshape a linear 3d system. **Right:** Reshaping that leads to loss of asymptotic stability. Here, an artificially generated planar circular trajectory centered at the origin is used with GP-MDS with a very large lengthscale.

tested trajectories did not converge to the origin, see Fig. 6.8b.

These are examples of particular systems with particular demonstrations and parameter settings that can not be used to draw any certain conclusions regarding asymptotic stability. What we can see though, is that unless repetitive behavior is explicitly demonstrated, the equilibrium point of the system seems to remain asymptotically stable. Planar systems are the only case in which it is easy to achieve closed orbits. With artificial data perfectly centered around the origin and with extreme parameter settings for the GP we were able to produce a system that was clearly not asymptotically stable. We conclude that in no case will orbital behavior occur unless the demonstrations explicitly include repetitive patterns, and even with such demonstrations the resulting system will only produce closed orbits in special cases. However, it is advisable to avoid repetitive patterns in the demonstrations, as they can lead to unnatural motions as in Fig. 6.7c.

6.4.3 POLISHING TASK USING PLANAR PERIODIC MOTION

From Section 6.4.2, it is clear that periodic motion is generally quite difficult to achieve with GP-MDS. In the special case of planar systems, however, periodic motion can be achieved by reshaping the dynamics into a limit cycle which can be half stable (Fig. 6.7a) or stable (Fig. 6.7b). Note that in planar systems, limit cycles can only occur if they encircle an equilibrium point. Since we consider with original dynamics that have a single stable equilibrium point, the location of the limit cycle is constrained to include this point. By reshaping at the origin, the latter can be turned from a sink to a source, resulting in system in which all trajectories converge to a limit cycle as in Fig. 6.7b. Note that to achieve this, Propositions 6.3 and 6.4 are violated and the origin is no longer stable.

Periodic motions that can be parameterized in the plane can hence be modeled using the proposed system. To exemplify this, we consider a robotic polishing task. The polishing task mimics the final brightening step of watches with a major Swiss watchmaker. This task is currently done manually at this company. Our implementation is a prototype meant to showcase a robotic system that could potentially ease the repetitive work of the polisher.

The motion is parameterized in 2d by one translational component and one rotational component. These were chosen as the z-coordinate and rotation around the y-axis in the reference frame of the polishing center. The remaining degrees of freedom remain constant during the task. As original dynamics, a linear system bringing the robot in a straight line to the polishing center was used. By starting the demonstration when the robot is on the equilibrium point of the DS, the latter is naturally destabilized. This effect can be seen in Fig. 6.9a, which shows the demonstrated data and the resulting reshaped system. When polishing objects of different shapes, sizes and materials, it is important to adapt the polishing motion accordingly. GP-MDS is a suitable modeling tool

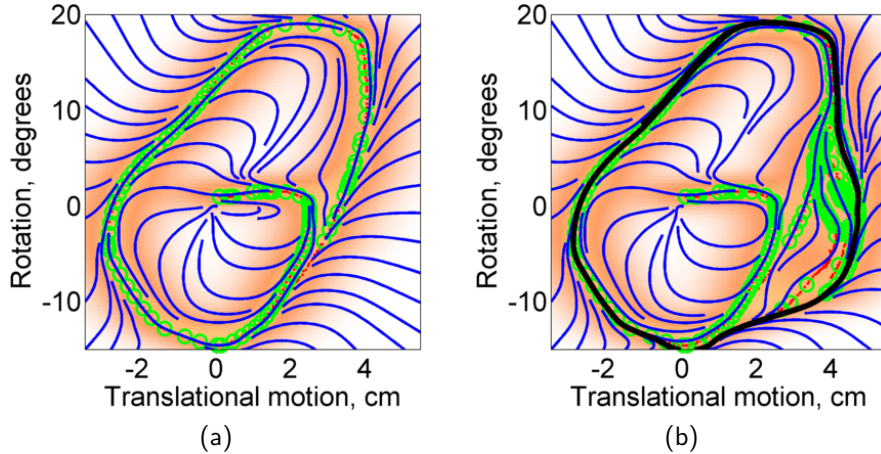


Figure 6.9: Left: The plot shows the initial demonstrated data and resulting limit cycle system. **Right:** For polishing of differently shaped objects, the polishing motion has to be adapted. Here, additional demonstrations were provided until a new satisfactory shape, highlighted by the black line, had been established.

for such tasks, since new data can be incorporated incrementally while retaining the periodic motion. The demonstrations seen in Fig. 6.9b were aimed at establishing a more pronounced orientation change before the work-piece is stroked in a linear motion over polishing wheel. Corrective data was provided until a satisfactory new limit cycle had been shaped.

6.4.4 CARTESIAN TRAJECTORY MODELING FOR STACKING PLATES

The task considered is to insert plates into slots in a dish rack, see Fig. 6.10b. To perform this task, the robot needs to grasp the plates and transport them from an arbitrary starting location to the slot, and insert them with the correct orientation. We focus on using the proposed methodology to learn a model for the translational motion of the end-effector. While general treatment of the grasping and orientation control aspects also present interesting problems per se, these are outside the scope of this work. We hence achieve proper orientation by keeping the end-effector orientation fixed. The grasping is completed by manual control of the Barrett Hand by a human operator.

As original dynamics, a Cartesian SEDS model corresponding to a standard place-type motion trained from trajectories recorded from humans was used. Example trajectories from this system are illustrated in Fig. 6.11a. As can be seen, the general motion pattern is appropriate for the task, but trajectories starting close to the dish rack tend to go straight toward the target, colliding with the rack on its path. This model can be improved by locally reshaping the

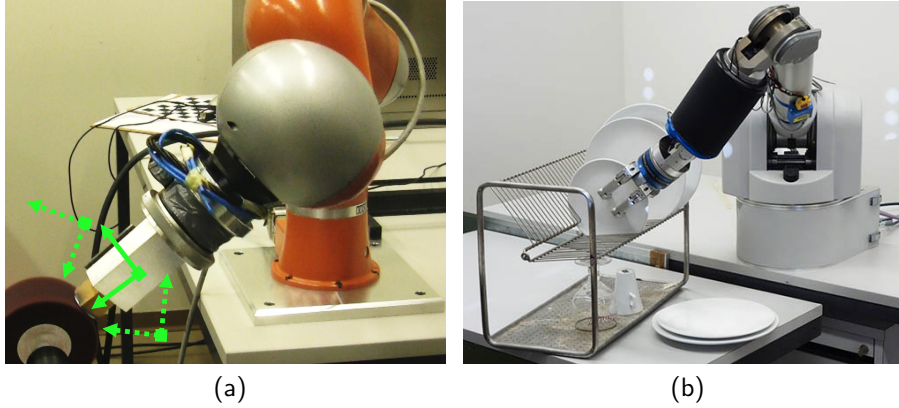


Figure 6.10: Left: The KUKA LWR robot performing the polishing task. The green arrows illustrate the planar polishing motion. **Right:** The Barrett WAM performing the Cartesian plate stacking task.

system in the problematic region.

For controlling the robot, we use a Cartesian impedance controller without inertia shaping (refer to Section 2.2.4) to follow the trajectory generated by the DS. The trajectory was integrated from the DS in the same way as for the pouring task described in Section 3.3.3. Since the teaching procedure takes place iteratively as the robot is performing the task, it is necessary to inform the system when a corrective demonstration is being delivered. We achieve this through the use of an artificial skin module mounted on the robot. The idea is to achieve accurate tracking in combination with compliant motion when necessary for corrective demonstrations. This is achieved by multiplying the feedback-component of the controller with a positive scalar which is inversely proportional to pressure detected on the artificial skin. Let $\tau_{PD} \in \mathbb{R}^7$ denote the vector of joint torques coming from the Cartesian impedance controller, $\tau_G \in \mathbb{R}^7$ denote the gravity compensation torques. The control torque τ commanded to the robot joints is then:

$$\tau = \psi \tau_{PD} + \tau_G \quad (6.13)$$

where $\psi \in [0, 1]$ is a truncated linear function which is equal to one when there is no pressure on the skin, and equal to zero when the detected pressure exceeds a predetermined threshold value. As an effect of Eq. (6.13), the resistance to perturbations is decreased when the teacher pushes the arm in order to deviate its trajectory during a corrective demonstration. Note that in contrast to Chapter 3, where the artificial skin was used to teach stiffness increase, the use here is just a practical means to combine accurate tracking with compliant motion when needed to give corrective demonstrations. This implementation was chosen as we focus in this chapter on the kinematic aspects of the task, and therefore wish to follow the DS trajectory with high accuracy.

The teaching procedure was initialized by starting the robot in a problematic

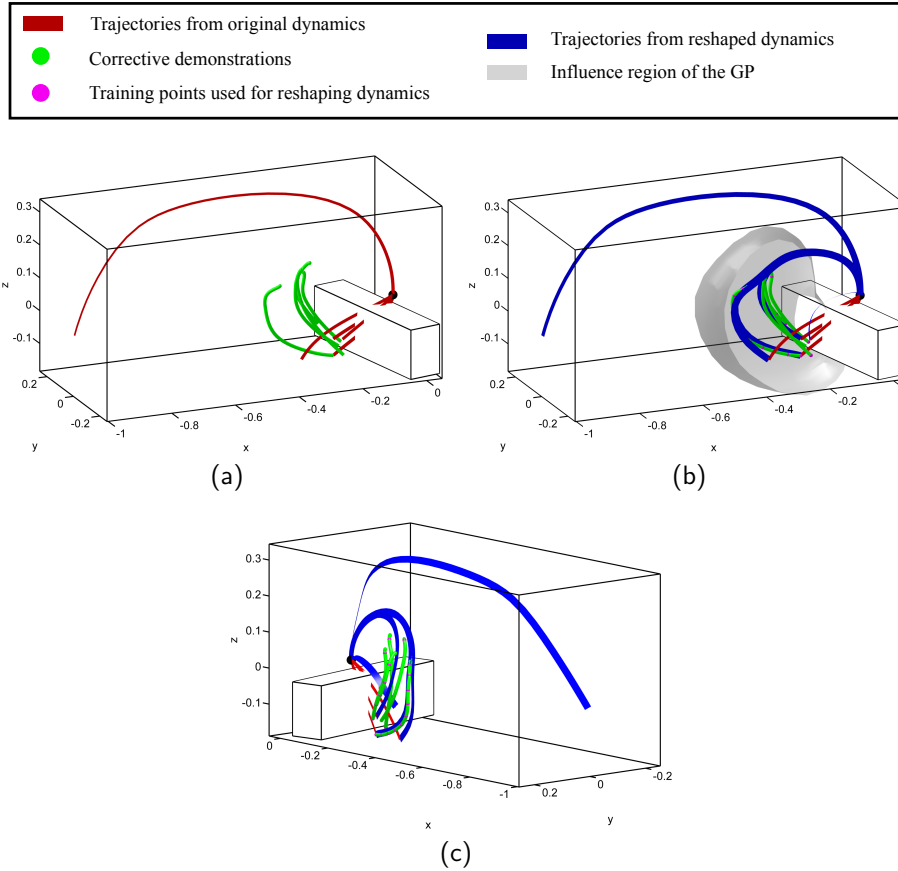


Figure 6.11: a) Trajectories resulting from the original dynamics from a set of starting points. Note that the overall motion seems well suited for the task, but trajectories starting close to the rack tend to collide with it. Provided corrective training data delivered through physical guiding of the robot is shown in green. b) Resulting reshaped system. The gray shaded region illustrates the region of influence of the GP and is computed as a level set of the predictive variance. c) Reshaped system from different point of view. Note the sparse selection of the training data.

starting point (one that would result in collision with the rack if the teacher does not intervene). The teacher then physically guided the robot during its motion so as to prevent collision with the rack. The data was recorded and used by GP-MDS to reshape the dynamics according to the incoming data. This procedure was repeated from a few problematic points in order to expand the reshaped region of the state space. Corrective demonstrations were provided during four trajectories starting in the problematic region, resulting in the training data visible in Figures 6.11b and 6.11c. A total of 1395 data points were collected. With lengthscale $l = 0.07$, signal variance $\sigma_f = 1$ and signal noise $\sigma_n = 0.4$ and with selection parameter values $\bar{J}^1 = 0.1, \bar{J}^2 = 0.2$, only 22 training points needed to be saved in the training set. The region in which the dynamics are reshaped is illustrated as a gray surface in Fig. 6.11b. As can be seen in Figures 6.11b and 6.11c, the dynamics were successfully reshaped to avoid collision with the edge of the rack. The total computational time (GP prediction followed by reshaping) took about 0.04ms, two orders of magnitude faster than required for our control frequency of 500Hz. The program was written in C++ and executed on an average desktop with a quad-core Intel Xeon processor. On this particular machine, the maximum number of training points compatible with our control frequency is just over 2000.

6.5 Discussion and conclusion

We have proposed a novel framework for incremental learning in dynamical systems. Desired behavior is achieved by locally applying transformations to an existing dynamical system. As such, this work is strongly related to previous work in our group (Khansari-Zadeh and Billard, 2012), which also uses full-rank modulations (but on a different form) in the context of dynamic obstacle avoidance. Here, we exploit such modulations not for obstacle avoidance but to learn the task itself.

The LMDS framework can be used with various forms of modulation function and learning algorithms. In this work, we have proposed a particular modulation function, based on locally scaling the speed and rotating the direction of the velocity. This modulation function proved very useful to locally make the streamlines of the DS match demonstrated trajectories. We would like to emphasize that this is one particular example of a possible modulation function, and a wealth of interesting behaviors could be implemented by a different choice of modulation function. For example, it would be straightforward to implement attraction or repulsion from an arbitrary axis in the work-space. The former could be very useful for locally encoding convergence of trajectories to a narrow path.

We proposed a particular algorithm, GP-MDS, which modulates the original dynamics by locally rotating and scaling it and learning the parameters of the modulation function with GPR. There exists numerous other regression tech-

niques that could be used instead of GPR, if desirable. The required property is that the regression must be local, so that the parameter vector approaches zero in regions remote from the demonstrations. Possible algorithms include Support Vector Regression (Drucker et al., 1997) which can achieve this by enforcing zero bias, Locally Weighted Learning (Atkeson et al., 1997) with radial basis functions and perhaps of particular interest for large data sets Locally Weighted Projection Regression (Vijayakumar et al., 2005).

The three hyper parameters of the GP (lengthscale, signal variance and noise variance) were selected by hand in this work. These parameters are easy to tune, and for a wide range of applications the signal and noise variance could be fixed and the lengthscale can be varied achieve the desired trade-off between generalization and local accuracy. Further open parameters are the thresholds J^1 and J^2 which determine the sparsity of the data used for GPR. Because of the form of the sparsity criteria, the thresholds represents quantities that have an interpretation in a trajectory context: they are the acceptable levels of error in speed and direction respectively. Note that there is no cap on the amount of data points used in GP-MDS, and problematic data set sizes could be reached despite the sparsity criteria if very rich and varying data is provided. This aspect could be improved by incorporating a maximum number of allowed training points and a matching strategy for pruning old data from the training set when necessary. Other directions for future research include Using a non-stationary covariance function, which would increase the flexibility by allowing the generalization/detail trade-off to vary across the state-space. Furthermore, using multiple GP:s with different lengthscales would potentially remove the need to compromise between generalization and local accuracy.

Generic methods that use task-based energy functions can allow incremental learning in DS, but these methods are always limited by the form of the task energy function. There are methods to deal with this difficulty, (Khansari-Zadeh and Billard, 2014; Lemme et al., 2014) which learns task-based energy functions based on the demonstrated data. These systems first build an estimate of a task-based Lyapunov function from demonstrations, and subsequently allows incremental adjustments only if they respect descent of the learned Lyapunov function. LMDS, which is not based on an energy function, in contrast supports unconstrained incremental learning.

Our experimentation indicates that with asymptotically stable original dynamics, systems reshaped with GP-MDS will retain convergence of all trajectories to the origin unless orbital behavior was explicitly demonstrated (Section 6.4.2). There are possible avenues for verifying the asymptotic stability property of a reshaped system. For planar systems, it is possible to conclude asymptotic stability by precluding the existence of limit cycles⁶. This can be done by invoking Bendixon’s criterion, which gives a sufficient condition for the non-existence of limit cycles depending on the Jacobian of the system (Slotine and Li, 1991).

⁶Limit cycles must enclose an equilibrium point

Generalizations of Bendixon’s criterion to higher dimensions, notably Smith’s autonomous convergence theorem (Smith, 1986) and extensions (Li and Muldowney, 1995), further allows to ensure asymptotic stability in any dimension. Depending on the form of the original dynamics, numerical evaluation across the state-space would generally be required to ensure asymptotic stability using such an approach.

In this chapter, we have focused on kinematic modeling of tasks. The system proposed here as well as any other DS representation relies on a controller to convert the output from the DS to motor commands for the robot. With conventional controllers, such as position control or impedance control, this requires integration of trajectories from the DS to yield a reference trajectory. Unfortunately, this practice is problematic for several reasons, which are detailed in the next chapter, where we also address this by introducing a new controller for execution of tasks described with a DS representation.

PASSIVE INTERACTION CONTROL WITH DYNAMICAL SYSTEMS

7.1 Introduction

Chapters 3 and 4 assumed the availability of a time-dependent trajectory describing the kinematic aspects of the task. The DS that was used in Section 3.3.3 was integrated open loop, without feedback from the robot state after the onset of motion. The same approach has also been used in previous works from our group (Kronander et al., 2011; Khansari-Zadeh and Billard, 2014, 2011; Kim et al., 2014). The open-loop integration results in a position trajectory which can hence be used in conventional control frameworks. Theoretically, trajectory generation with DS guarantees robustness in the sense that the trajectory is generated online and hence allows to take any unplanned perturbations into account. When the DS is integrated open-loop as described in Section 3.3.3, part of this robustness is lost. While the robustness with respect to perturbations such as moving target point can still be ensured, perturbations on the robot itself cannot be handled since the DS is agnostic to the actual state of the robot, see Fig. 7.1. This chapter is entirely devoted to the development of a control method that allow to make full use of the DS, specifically to allow feedback of the actual robot state to the DS, hence allowing the DS to handle *intrinsic* as well as *extrinsic* perturbations.

An important property of impedance control is the passivity property, which guarantees not only asymptotically stable regulation in the absence of perturbations, but also stable interaction with any passive environment. It is important to note that the latter is true when passivity is ensured with *external force as input and velocity as output*. From now on, when we refer to passivity without specifying the input and output, it should be understood as passivity with respect to external force as input and robot velocity as output. In this sense, classic impedance control is only passive in the regulation case (Section 2.2.4), and the passivity property is lost if the desired velocity is non-zero. The loss of passivity during tracking is a drawback of impedance control and a problem that arises in any controller driven by time-indexed reference trajectories. While many passivity based trajectory tracking controllers have been developed, e.g. (Slotine, 1987), these works typically ensure passive mapping from external force to a tracking error variable. As has been pointed out e.g. in (Li and Horowitz,

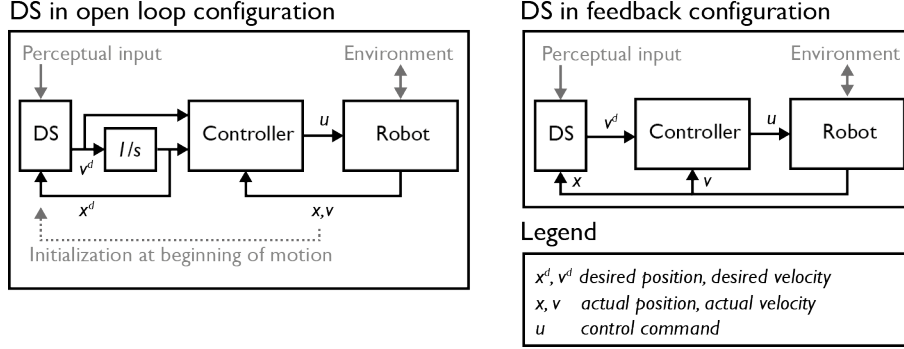


Figure 7.1: Illustration of open loop and closed loop control configurations with DS. In the open loop configuration, the DS is updated with the desired position resulting from integration of the desired velocity. The actual position of the robot is only used for initializing the integration at the beginning of the motion. In contrast, the feedback configuration continuously updates the DS with the actual position and realizes a control on the velocity error.

1999; Duindam et al., 2004), that passivity mapping can ensure stability in free motion *but not in contact*. In this work, it is shown how this problem can be remedied by replacing the reference trajectory by an DS that defines the desired velocity as a function only of the current state of the robot. We propose a passive DS controller which is based on a variable damping matrix designed to dissipate selectively in directions perpendicular to the desired direction of motion. Tuning the singular values of this damping matrix allows to vary the dynamic relationship between external force and velocity (impedance). The application of such a damping matrix results in a system whose natural behavior is to follow the streamlines of the DS describing the task. Driving control is then provided by exploiting a decomposition of the DS into a conservative and non-conservative part, and by introducing an auxiliary state variable that can temporarily store a bounded amount of energy.

7.2 Problem Statement

Let $\mathbf{f}(\boldsymbol{\xi})$ be a Dynamical System describing the nominal motion plan for a robotic task. The variable $\boldsymbol{\xi}$ represents a generalized state variable, which could be e.g. robot joint angles or Cartesian position. Any integral curve of \mathbf{f} represents the desired motion of the robot in the absence of perturbations. Consider a dynamics of a RBD with the generalized state variable $\boldsymbol{\xi}$:

$$\mathbf{M}(\boldsymbol{\xi})\ddot{\boldsymbol{\xi}} + \mathbf{C}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})\dot{\boldsymbol{\xi}} + \mathbf{g}(\boldsymbol{\xi}) = \boldsymbol{\tau}_c + \boldsymbol{\tau}_e \quad (7.1)$$

As in Section 2.2.3, $\mathbf{M}(\boldsymbol{\xi})$ and $\mathbf{C}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})$ denote the state dependent inertia and centrifugal/Coriolis matrices. The goal of this work is to design a controller $\boldsymbol{\tau}_c$ guiding the system along the nominal motion specified by a given $\mathbf{f}(\boldsymbol{\xi})$ so that

Eq. (7.1) has the following properties:

1. Passivity $(\tau_e, \dot{\xi})$ should be preserved for the controlled system.
2. The controller should dissipate kinetic energy in directions not relevant for the task.
3. It should be possible to vary task-based impedance of the manipulator, e.g. how dynamics defining how external forces τ_e affect the velocity $\dot{\xi}$.

7.3 Proposed Approach

7.3.1 SELECTIVE ENERGY DISSIPATION WITH TASK VARYING DAMPING

Consider a feedback controller consisting solely of a damping term and a gravity cancellation term:

$$\tau_c = \mathbf{g}(\xi) - \mathbf{D}\dot{\xi} \quad (7.2)$$

where $\mathbf{D} \in \mathbb{R}^{N \times N}$ is some positive semi-definite matrix. Obviously, since the nominal model $\mathbf{f}(\xi)$ does not appear in Eq. (7.2), we do not expect that the system would converge to the desired velocity under this simple control. However, as we shall see we are able to address other design goals listed in Section 7.2.

It is easy to show that the controller in (7.2) renders the system (7.1) passive with respect to the input τ_e , output $\dot{\xi}$ with the kinetic energy as storage function. This is true for an arbitrarily varying damping, as long as it remains positive semi-definite. We will exploit this fact and construct a varying damping term that dissipates selectively in directions orthogonal to the desired direction of motion given by $\mathbf{f}(\xi)$. Let $\mathbf{e}_1, \dots, \mathbf{e}_N$ be an orthonormal basis for \mathbb{R}^N with \mathbf{e}_1 pointing in the desired direction of motion. Hence, let $\mathbf{e}_1 = \frac{\mathbf{f}(\xi)}{\|\mathbf{f}(\xi)\|}$, and let $\mathbf{e}_2, \dots, \mathbf{e}_N$ be an arbitrary set of mutually orthogonal and normalized vectors. Let the matrix $\mathbf{Q}(\xi) \in \mathbb{R}^{N \times N}$ be a matrix whose columns are $\mathbf{e}_1, \dots, \mathbf{e}_N$. This matrix is a function of the state ξ , since the vectors \mathbf{e}_1 and hence all $\mathbf{e}_1, \dots, \mathbf{e}_N$ depend on ξ via $\mathbf{f}(\xi)$. We then define the state-varying damping matrix $\mathbf{D}(\xi)$ as follows:

$$\mathbf{D}(\xi) = \mathbf{Q}(\xi)\mathbf{\Lambda}\mathbf{Q}(\xi)^T \quad (7.3)$$

where $\mathbf{\Lambda}$ is a diagonal matrix with non-negative values on the diagonal $\lambda_1, \dots, \lambda_N \geq 0$.

By adjusting these damping values, different dissipation behaviors can be achieved. For example, setting $\lambda_1 = 0$ and $\lambda_2, \dots, \lambda_N > 0$ results in a system that selectively dissipates energy in directions perpendicular to the desired mo-

tion. Hence, external work in irrelevant directions is opposed while along the integral curves of $\mathbf{f}(\boldsymbol{\xi})$ the system is free to move.

7.3.2 TRACKING IN CONSERVATIVE DS

While the selective damping in Section 7.3.1 allowed selective energy dissipation, it can not drive the robot forward along the integral curves of \mathbf{f} . In order to achieve this, we have to add some driving control to Eq. (7.2). This can be achieved through rather simple means, provided that the nominal task model \mathbf{f} is the negative gradient of an associate potential function. This is a restricted class of DS will be referred to as conservative vector fields in the remainder of this chapter.

Definition 7.1 (Conservative DS). An autonomous dynamical system $\mathbf{f}(\boldsymbol{\xi})$ is conservative if and only if there exists a scalar potential function such that:

$$\mathbf{f}(\boldsymbol{\xi}) = -\nabla V_{\mathbf{f}}(\boldsymbol{\xi}) \quad (7.4)$$

Consider now a modified controller with negative velocity error feedback:

$$\boldsymbol{\tau}_c = \mathbf{g}(\boldsymbol{\xi}) - \mathbf{D}(\boldsymbol{\xi})(\dot{\boldsymbol{\xi}} - \mathbf{f}(\boldsymbol{\xi})) = \mathbf{g}(\boldsymbol{\xi}) - \mathbf{D}(\boldsymbol{\xi})\dot{\boldsymbol{\xi}} + \lambda_1 \mathbf{f}(\boldsymbol{\xi}) \quad (7.5)$$

The last equality is due to the fact that $\mathbf{f}(\boldsymbol{\xi})$ is an eigenvector of $\mathbf{D}(\boldsymbol{\xi})$ as described in Section 7.3.1.

Proposition 7.1. *Let $\mathbf{f}(\boldsymbol{\xi})$ be a conservative system with an associated potential function $V_{\mathbf{f}}(\boldsymbol{\xi})$. Then, the system (7.1) under control given by (7.5) is passive with respect to the input output pair $\boldsymbol{\tau}_e, \dot{\boldsymbol{\xi}}$ with the storage function $W(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \frac{1}{2} \dot{\boldsymbol{\xi}}^T \mathbf{M}(\boldsymbol{\xi}) \dot{\boldsymbol{\xi}} + \lambda_1 V_{\mathbf{f}}(\boldsymbol{\xi})$*

Proof. The proof is given in Appendix C.2. □

It is important to remark that it is never necessary to evaluate the potential function, what is important is its existence and that its negative gradient is given by $\mathbf{f}(\boldsymbol{\xi})$. The controller derived in this section is strongly related to a class of controllers based on energy-shaping, pioneered by (Takegaki and Arimoto, 1981). If all DS of interest were conservative vector fields, the controller in Eq. (7.5) would already satisfy at least two points from the list of desired properties in Section 7.2. Unfortunately, most interesting DS models are not conservative. In particular, LMDS systems (Chapter 6) are generally not conservative.

7.3.3 EXTENSION TO NON-CONSERVATIVE DS

The restriction to conservative systems previously was necessary to cancel the term $\lambda_1 \dot{\boldsymbol{\xi}}^T \mathbf{f}(\boldsymbol{\xi})$ in the rate of change of the kinetic energy in the system. However,

that term only has to be canceled for $\dot{\boldsymbol{\xi}}^T \mathbf{f}(\boldsymbol{\xi}) > 0$. If $\dot{\boldsymbol{\xi}}^T \mathbf{f}(\boldsymbol{\xi}) \leq 0$, the control is contributing to the decrease of kinetic energy. Consider at time t_0 that the system has a particular energy level $W(t_0)$. Now assume that at $t_1 > t_0$ energy has been dissipated so that $W(t_1) < W(t_0)$. This dissipated energy provides in a sense a passivity margin — the system would be passive even if $W(t_1) = W(t_0)$. This fact can be exploited by augmenting the state vector with a virtual state that is capable of storing energy that would otherwise be lost in dissipation. This stored energy can then be released in order to implement control actions that would be non-passive in the original system without the storage element. This concept, sometimes referred to as energy tanks, feature prominently bilateral tele-manipulation (Franken et al., 2011; Lee and Huang, 2010; Stramigioli et al., 2005) and has also been applied to variable stiffness control (Ferraguti et al., 2013) which is discussed in detail in Chapter 4.

Let $\mathbf{f}(\boldsymbol{\xi})$ be decomposed into a conservative part and a non-conservative part:

$$\mathbf{f}(\boldsymbol{\xi}) = \mathbf{f}_C(\boldsymbol{\xi}) + \mathbf{f}_R(\boldsymbol{\xi}) \quad (7.6)$$

where \mathbf{f}_c denotes the conservative part which has an associated potential function, and \mathbf{f}_R denotes the non-conservative part. Note that any system can be written on this form, e.g. for divergence-free systems $\mathbf{f}_c \equiv 0$. In particular, a LMDS system (Chapter 6) have a known decomposition if *the original dynamics is a conservative vector field*. In the following, the symbol V_C will be used to denote the potential function associated to the conservative part of the DS.

We shall consider an additional state variable $s \in \mathbb{R}$ that represents stored energy. It is a virtual state to which we can assign arbitrary dynamics. We shall consider dynamics coupled with the robot state variables $\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}$ as follows:

$$\dot{s} = \alpha(s) \dot{\boldsymbol{\xi}}^T \mathbf{D} \dot{\boldsymbol{\xi}} - \beta_s(z, s) \lambda_1 z \quad (7.7)$$

where $z = \dot{\boldsymbol{\xi}}^T \mathbf{f}_R(\boldsymbol{\xi})$. The scalar functions $\alpha : \mathbb{R} \mapsto \mathbb{R}$ and $\beta : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ control the flow of energy between the virtual storage s and the robot, and will be defined in the following. It is necessary to put an upper bound on the virtual storage, such that it can only store a finite amount of energy. Let $\bar{s} > 0$ denote this upper bound. Then, $\alpha(s)$ should satisfy:

$$\begin{cases} 0 \leq \alpha(s) \leq 1 & s < \bar{s} \\ \alpha(s) = 0 & s \geq \bar{s} \end{cases} \quad (7.8)$$

Disregarding for the moment the second term in Eq. (7.7), it is clear that the first term (energy that would otherwise be dissipated) only adds to the virtual storage as long as the latter remains below its upper bound, $s < \bar{s}$. Now turning

to the second term of Eq. (7.7), $\beta_s(z, s)$ should satisfy:

$$\begin{cases} \beta_s(z, s) = 0 & s \leq 0 \text{ and } z \geq 0 \\ \beta_s(z, s) = 0 & s \geq \bar{s} \text{ and } z \leq 0 \\ 0 \leq \beta_s(z, s) \leq 1 & \text{elsewhere} \end{cases} \quad (7.9)$$

Considering the second term in Eq. (7.7), it is clear that with β_s satisfying Eq. (7.9), transfer to the virtual storage ($z < 0$) is only possible as long as $s < \bar{s}$. Conversely, extraction of energy from the storage ($z > 0$) is only possible as long as $s > 0$. When the storage is depleted, the controller can no longer be allowed to drive the system along \mathbf{f}_R if this results in increasing the kinetic energy of the system. Therefore, we introduce the scalar function $\beta_R(z, s)$ whose role is to modify the control signal if the storage is depleted.

$$\boldsymbol{\tau}_c = \mathbf{g}(\boldsymbol{\xi}) - \mathbf{D}\dot{\boldsymbol{\xi}} + \lambda_1 \mathbf{f}_c(\boldsymbol{\xi}) + \beta_R(z, s) \lambda_1 \mathbf{f}_R(\boldsymbol{\xi}) \quad (7.10)$$

where $\beta_R : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ is a scalar function that should satisfy:

$$\begin{cases} \beta_R(z, s) = \beta_s(z, s) & z \geq 0 \\ \beta_R(z, s) \geq \beta_s(z, s) & z < 0 \end{cases} \quad (7.11)$$

We are now ready to state the main result of this section:

Theorem 7.1. *Let the nominal task model $\mathbf{f}(\boldsymbol{\xi})$ be composed of conservative and non-conservative parts according to Eq. (7.6). Let the system (7.1) be controlled by Eq. (7.10) and assume the functions α, β_s, β_R satisfy the conditions in Equations (7.8), (7.9) and (7.11) respectively. Let $0 < s(0) \leq \bar{s}$. The resulting closed loop system is passive with respect to the input-output pair $\boldsymbol{\tau}_e, \dot{\boldsymbol{\xi}}$.*

Proof. The proof is given in Appendix C.3. □

The specifications of the functions α, β_s, β_R allow some freedom in the design. In this work, we use compositions of smooth step functions as detailed in appendix C.1.

7.3.4 IMPEDANCE ADJUSTMENT

The architecture used in this chapter differs fundamentally from the classical impedance control framework in that there is no notion of reference position. Instead, there is only a reference velocity, which is generated online as a function of the robot position. While in general, an impedance is by no means restricted to the classical spring-damper relationship, it is useful to analyze the link between the impedance rendered by the proposed controller and the classical spring-damper setting.

To aid the discussion, we write the closed loop dynamics of the system (7.1) under control $\boldsymbol{\tau}_e = \mathbf{g}(\boldsymbol{\xi}) - \mathbf{D}\dot{\boldsymbol{\xi}} + \lambda_1 \mathbf{f}(\boldsymbol{\xi})$. Note that the passive controller derived in Section 7.3.3 yields the same closed loop behavior in the ideal case that the virtual storage is never depleted.

$$\mathbf{M}(\boldsymbol{\xi})\ddot{\boldsymbol{\xi}} + (\mathbf{D}(\boldsymbol{\xi}) + \mathbf{C}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}))\dot{\boldsymbol{\xi}} - \lambda_1 \mathbf{f}(\boldsymbol{\xi}) = \boldsymbol{\tau}_e \quad (7.12)$$

Hence, as is the case of the 'simple' impedance controllers without inertia shaping in Section 2.2.4, we are unable to alter the inherent inertia of the robot. We have control of the damping in directions orthogonal to the desired motion via the damping values $\lambda_2 \dots \lambda_N$ which are allowed to vary with time, state or any other variable (see Section 7.3.1). The stiffness term is replaced by $\lambda_1 \mathbf{f}(\boldsymbol{\xi})$ which can be interpreted as a nonlinear stiffness term. This interpretation is evident when considering the behavior of Eq. (7.12) close to a stable equilibrium point of \mathbf{f} . For simplicity, consider the robot in steady state ($\dot{\boldsymbol{\xi}} = \ddot{\boldsymbol{\xi}} = \mathbf{0}$) near an equilibrium point $\boldsymbol{\xi}^*$ such that $\mathbf{f}(\boldsymbol{\xi}^*) = \mathbf{0}$. Accounting for steady state and approximating the left-hand side of Eq. (7.12) with a first order Taylor expansion around $\boldsymbol{\xi}^*$ then yields:

$$-\lambda_1 \left. \frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}} \right|_{\boldsymbol{\xi}=\boldsymbol{\xi}^*} (\boldsymbol{\xi} - \boldsymbol{\xi}^*) = \boldsymbol{\tau}_e \quad (7.13)$$

corresponding to a steady-state stiffness equal to the Jacobian of \mathbf{f} at the equilibrium point scaled by the value of λ_1 . Globally, the term $\lambda_1 \mathbf{f}(\boldsymbol{\xi})$ can be interpreted as a nonlinear stiffness term *centered on the equilibrium point(s) of \mathbf{f}* .

While the classical notion of stiffness manifests itself in vicinity of the equilibrium points of \mathbf{f} , it is not generally possible to generalize this to stiffness around general points in the work-space. To see this, consider again the steady-state linearization of the left-hand side of Eq. (7.12), but this time around an arbitrary point $\boldsymbol{\xi}'$ with $\mathbf{f}(\boldsymbol{\xi}') \neq \mathbf{0}$:

$$-\lambda_1 \mathbf{f}(\boldsymbol{\xi}') - \lambda_1 \left. \frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}} \right|_{\boldsymbol{\xi}=\boldsymbol{\xi}'} (\boldsymbol{\xi} - \boldsymbol{\xi}') = \boldsymbol{\tau}_e \quad (7.14)$$

A key observation is that a 'stiffness behavior' includes symmetry, where perturbations around a point on the desired trajectory are opposed uniformly around the reference trajectory. The DS task model, on the other hand, encodes infinitely many desired trajectories, given by the integral curves of \mathbf{f} . Hence, if the classical behavior of symmetrically converging toward a *fixed* trajectory is desired, this should be *encoded in the task model \mathbf{f}* . An example of a DS that locally encodes this spring-like behavior is given in Fig. 7.2.

An interesting perspective of the proposed controller is that the *resistance* to a perturbation can be tuned independently of the *recovery* from a perturbation. In impedance control, both of these are essentially determined by the stiffness

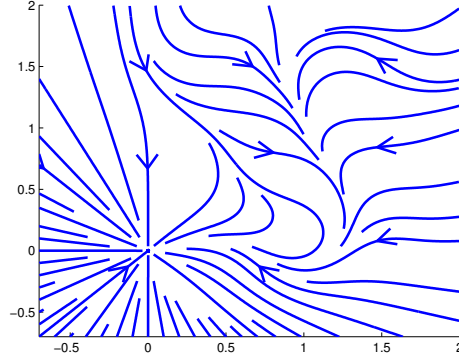


Figure 7.2: A 2d illustration of a task model that will locally yield stiffness in the classical sense around a fixed trajectory.

term and are hence identical: if a force tries to push the robot off its reference trajectory it will be opposed, and when the perturbing force is released the robot will return to the reference trajectory. This behavior is just a special case of what is possible with the proposed controller. For example, the proposed controller would also oppose a force that moves the robot perpendicular to its desired velocity. But after the force is released, the robot may resume the task along a different path.

7.3.5 CHOOSING \bar{s}

In this chapter, we assume that there is a DS available that represents the motion of a robotic task. If we furthermore assume that the desired parameters $\lambda_1, \dots, \lambda_D$ are available, there is one open parameter left that is crucial for correct operation of the controller. Choosing \bar{s} too small will result in poor tracking performance in strongly non-conservative systems, since the system will have to fall back on the conservative part of the DS when the virtual storage is depleted. Conversely, allowing storage of a high amount of energy may be a safety concern. In this section, we provide a simple heuristic that can be used to get an estimate of an appropriate \bar{s} for being able to follow a given integral curve of the DS.

Consider the case of perfect following of the desired DS (no external forces). The role of the virtual storage is to ensure energy balancing, so that for all $t_2 > t_1$:

$$U(t_2) + s(t_2) \leq U(t_1) + s(t_1) \quad (7.15)$$

where

$$U(t) = V_K^d(t) + \lambda_1 V_C(t) \quad (7.16)$$

Where $V_C(t)$ denotes the potential function associated with the conservative part of the DS, \mathbf{f}_C . Here, we have introduced the desired kinetic energy, defined as:

$$V_K^d(t) = \frac{1}{2} \mathbf{f}(\boldsymbol{\xi})^T \mathbf{M}(\boldsymbol{\xi}) \mathbf{f}(\boldsymbol{\xi}) \quad (7.17)$$

Regrouping the difference in virtual storage on the right-hand side, we get:

$$U(t_2) - U(t_1) \leq s(t_1) - s(t_2) \quad (7.18)$$

Note that unless $U(t)$ has a local maximum at $t > 0$ there is no need for any virtual storage at all. Therefore, we can proceed by identifying the largest local maximum of U , for $t > 0$. Therefore, now let t_2 denote the time of the largest local maxima of U with $t_2 > 0$. Assuming that t_2 has been identified, we set \bar{s} as:

$$\bar{s} = U(t_2) - \min_{0 < t < t_2} U(t) \quad (7.19)$$

To evaluate this expression we just need to be able to compute U . To simplify this, we introduce an upper-bound for the desired kinetic energy:

$$\hat{V}_K^d(t) = \frac{m}{2} \mathbf{f}(\boldsymbol{\xi})^T \mathbf{f}(\boldsymbol{\xi}) \quad (7.20)$$

where m is a scalar that is larger than the maximum eigenvalue of $\mathbf{M}(\boldsymbol{\xi})$ over all possible configurations. Let $\mathbf{c}(t)$ be a solution to $\dot{\boldsymbol{\xi}} = \mathbf{f}(\boldsymbol{\xi})$ with some given initial state $\boldsymbol{\xi}_0$. The solution curve is given by integrating the desired DS either for a predetermined time or until an attractive equilibrium point has been reached. We can immediately compute \hat{V}_K^d along the solution trajectory $\mathbf{c}(t)$. To compute the second term of Eq. (7.16), we need not be able to compute $V_C(t)$ directly as it can be integrated from the conservative part of the dynamics:

$$V_C(t) = - \int_{\mathbf{c}} \mathbf{f}_C(\boldsymbol{\xi}) \cdot d\mathbf{c} + V_C(0) = - \int_0^t \mathbf{f}_C(\boldsymbol{\xi}(t))^T \mathbf{f}(\boldsymbol{\xi}(t)) dt + V_C(0) \quad (7.21)$$

Excluding the offset term $V_C(0)$ which affects neither location of the extrema nor Eq. (7.19) we can easily evaluate this integral numerically.

Computing \bar{s} as described here is merely a heuristic that can be used to get a rough estimate of an appropriate value. Neither the robot dynamics nor the environment interaction has been taken into account, so there can be no guarantee that the computed value will be sufficiently high for the task. We have found, however, that this method works well in simulation 7.4.1 as well as on our real implementation

7.4 Simulations

This section presents a series of simulations aimed at highlighting various aspects of the proposed controller.

7.4.1 PLANAR FREE MOTION

This simulation is aimed at illustrating how the virtual storage is affected by the DS and accuracy of control. We consider a two-link robot, without friction and with link length of 1.2m and a mass of 3kg evenly distributed in the link.

The Robotics Toolbox (Corke, 2011) was used to simulate the dynamics of this robot.

TASK DESCRIPTION AND SIMULATED CONTROLLERS

The DS is GP-MDS model, with an encoded curve as can be seen in Fig. 7.3. We considered two setups:

- A.** Standard PICDS according to Eq. (7.10)
- B.** An additional nominal control term, which has the effect of conserving kinetic energy in directions aligned with the desired DS:

$$\boldsymbol{\tau}_c = \mathbf{g}(\boldsymbol{\xi}) - \mathbf{D}(\boldsymbol{\xi})\dot{\boldsymbol{\xi}} + \lambda_1 \mathbf{f}_c(\boldsymbol{\xi}) + \beta_R(z, s) \lambda_1 \mathbf{f}_R(\boldsymbol{\xi}) + \boldsymbol{\tau}_n \quad (7.22)$$

where the nominal control is given by:

$$\boldsymbol{\tau}_n = \frac{\langle \dot{\boldsymbol{\xi}}, \dot{\boldsymbol{\xi}}_r \rangle^2}{\langle \dot{\boldsymbol{\xi}}_r, \dot{\boldsymbol{\xi}}_r \rangle^2} \mathbf{M}(\boldsymbol{\xi}) (\langle \dot{\boldsymbol{\xi}}, \dot{\boldsymbol{\xi}}_r \rangle \ddot{\boldsymbol{\xi}}_d - \langle \dot{\boldsymbol{\xi}}, \ddot{\boldsymbol{\xi}}_d \rangle \dot{\boldsymbol{\xi}}_d) \quad (7.23)$$

where the inner product is with respect to the metric defined by $\mathbf{M}(\boldsymbol{\xi})$, $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T \mathbf{M}(\boldsymbol{\xi}) \mathbf{b}$. This command was proposed in (Duindam et al., 2004), and has the effect of keeping kinetic energy separated in desired and undesired directions. It can be shown that $\boldsymbol{\tau}_n$ is power-continuous, i.e. unlike other inverse dynamics approaches such as computed torque control it does not destroy the passivity of the system. Importantly, $\boldsymbol{\tau}_n$ is power continuous even in the presence of model error (incorrect \mathbf{M}). In this case, the performance will suffer *but not passivity*.

Both controllers were started at the same joint configuration (see Fig. 7.3), and 5 seconds were simulated. In both cases, $\lambda_1 = 5$ and $\lambda_2 = 30$ were used as damping parameters.

RESULTS

At the beginning of the trajectory, the robot is moving in a direction roughly aligned with the linear original dynamics, see Fig. 7.3. This results in the associated potential function initially decreasing (Fig. 7.4b, bottom). Around one second into the simulation, the robot reaches the reshaped region and starts to move in an increasing direction of the quadratic potential function. To keep the sum of kinetic, potential and virtual constant, the virtual energy therefore has to decrease. This can be seen in Fig. 7.4b, top.

As expected, the tracking performance is clearly better when the nominal control is active (Simulation B). This can be seen in the integrated trajectory (Fig. 7.3) as well as the velocity error (Fig. 7.4a). Note especially the sharp increase in velocity error around 3.5 seconds into the simulation. This happens as the robot should 'U-turn' after moving against the original dynamics. Using

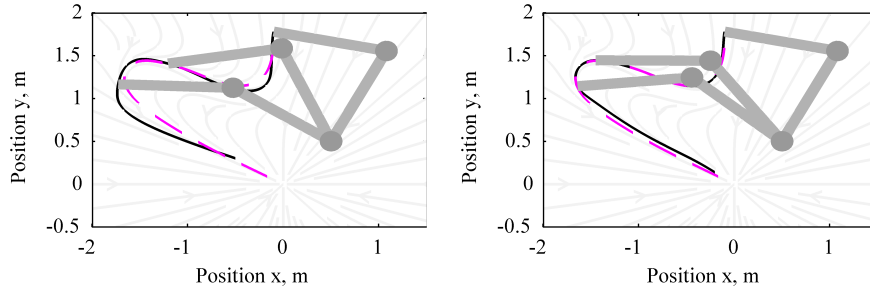


Figure 7.3: Open loop integrated trajectory (magenta dashed) and actual followed trajectory (black solid) for the two simulations. The two-link robot is shown at a series of points along the trajectory. **Left:** Simulation A, without nominal control. **Right:** Simulation B, with nominal control.

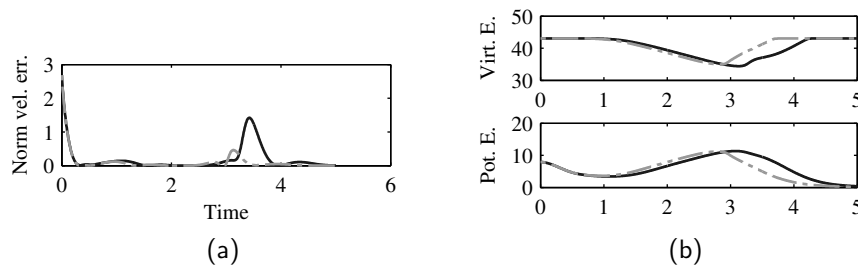


Figure 7.4: **Left:** Norm of velocity error for the simulations. As expected, better performance is achieved using the nominal control. **Right:** Virtual energy storage and potential function of original dynamics over time. As the potential function increases, the virtual storage decreases so that their sum with the kinetic energy remains non-increasing.

the nominal control here allows reuse the built up kinetic energy, while simulation A has to dissipate all the that energy and at the same time accelerate the robot in the new direction. This results in a big local velocity error which when integrated results in a significant departure from the open-loop trajectory (see Fig. 7.3, left).

7.4.2 BEHAVIOR IN UNEXPECTED CONTACT

An important advantage of the proposed control scheme with a DS and a feedback velocity command is that there will be no temporal ramp-up of the contact force in the event of an unexpected obstacle (or human) interfering with the movement. If such an event occurs with PICDS, the contact force is determined by the DS and the value of λ_1 . Depending on these, the contact force may be high but it will remain constant. We believe this is a crucial feature for safe HRI.

The DMP framework was designed to learn and generate desired trajectories. For converting the trajectories into control commands, it relies on a lower level controller, e.g. a position controller or impedance controller. In its standard

formulation, the phase variable of the DMP is an autonomous system acting as a clock for the motion. Hence, the DMP appears susceptible to the same type of problems as a time-dependent reference trajectory and impedance controller. However, it was proposed already in (Ijspeert et al., 2002) and recently reviewed in (Ijspeert et al., 2013) to introduce coupling control in the DMP, affecting the transformation dynamics, canonical dynamics or both. As recently emphasized in (Gams et al., 2014) however, when such coupling is used the guarantee of stability of the DMP is lost. Despite this, several works have demonstrated that coupling terms can be used e.g. to adapt motion according to an expected sensation during a task (Pastor et al., 2011), cooperative manipulation (Kulvicius et al., 2013) and reference adaptation for accurate force control through iterative learning control (Gams et al., 2014). These works make use of force feedback in the coupling terms. Coupling from the position error as exemplified in (Ijspeert et al., 2013) is a mechanism most relevant in comparison to feedback of robot position in PICDS. We conducted a simulation to elucidate the differences between these approaches.

TASK DESCRIPTION

As in (Ijspeert et al., 2013), we consider a uni-dimensional task for illustrating the behavior in unexpected contact. A fifth order polynomial fitted to have zero first and second derivatives at $t=0$ and $t=1$ is used as a reference trajectory. Let $x^r(t) \in \mathbb{R}$ denote the reference trajectory, defined as:

$$x^r = \begin{cases} 6t^5 - 15t^4 + 10t^3 & 0 \leq t < 1 \\ 1 & t \geq 1 \end{cases} \quad (7.24)$$

Since this is monotonic function for $t \geq 0$, the derivative \dot{x}^r can be written as a function of x , $\dot{x}^r = f(x)$.

When we simulate the system, the actual velocity is artificially frozen in the time interval $[0.3, 0.9]$. With the exception of the reference trajectory, which we have slightly modified to make it monotonic so that it can be described as an autonomous DS in the uni-dimensional case, these simulation conditions are identical to the example for position error coupling in (Ijspeert et al., 2013).

CONTROLLERS

Three setups are simulated. The first uses the time-dependent reference trajectory and PD control law:

- A.** Time-dependent reference trajectory and PD control:

$$\ddot{x} = -K(x - x^r) - D(\dot{x} - \dot{x}^r) \quad (7.25)$$

with $K = 1000$ and $D = 2\sqrt{K}$.

B. DMP with coupling terms:

Canonical phase phase system:

$$\tau \dot{s} = -\alpha_s s \quad (7.26a)$$

Forcing term:

$$\phi = \frac{\sum_{n=1}^{20} \psi_n(s) w_n}{\sum_{n=1}^{20} \psi_n(s)} s \quad (7.26b)$$

$$\psi_n(s) = \exp(-h(s - c_n)^2)$$

where $h = 0.01$ and $c_1 \dots c_{20}$ are distributed uniformly in $[0, 1]$.

Transformation system:

$$\tau z = \alpha_z (\beta_z (g - x^r) - z) + \phi + C_T \quad (7.26f)$$

$$\tau \dot{x}^r = z \quad (7.26g)$$

Position error filter:

$$\dot{e} = \alpha_e (x - x^r - e) \quad (7.26c)$$

Coupling terms:

$$C_T = k_T e \quad (7.26d)$$

$$\tau = 1 + k_C e^2 \quad (7.26e)$$

To fit the open parameters of the DMP, $x^r, \dot{x}^r, \ddot{x}^r$ were simulated for $t \in [0, 1]$ and used as training data for a DMP with 20 receptive fields. The reference is then taken from the DMP and tracked identically as Eq. (7.25). The open coupling gains were set to $k_T = 10^3$ and the time constant coupling gain $k_C = 10^4$. The setup with exception for the training data provided to the DMP is identical to the coupling example in (Ijspeert et al., 2013).

C. The proposed controller:

$$\ddot{x} = -\lambda_1 (\dot{x} - f(x)) \quad (7.27)$$

with $\lambda_1 = D$ for fairness of comparison with (7.25).

RESULTS

The resulting position, velocity and force traces for all three simulations are shown in Fig. 7.6. As expected, setup A with a time-dependent trajectory yields an increasing contact force. A violent conversion from potential to kinetic energy takes place at the end of the blocked period. It is easy to see the adaptation of the reference trajectory for setup B, and how this affects the contact force, which remains quasi-stable at a significantly lower value than setup A. Finally setup C has again a lower contact force which remains stable during the contact.

This is an illustrative example that is not show all effects that would occur in a real scenario. Nonetheless, it shows that timed trajectory controllers can yield hazardous behavior in situations where unexpected contact may occur. Secondly, the behavior of DMP with coupling terms is similar to the PICDS while needing considerably more design effort.

Indeed, the coupling parameters in DMP must be chosen carefully. Fig. 7.6c plots simulation of setup B but where the gains are changed to $k_T = 10^4$ and

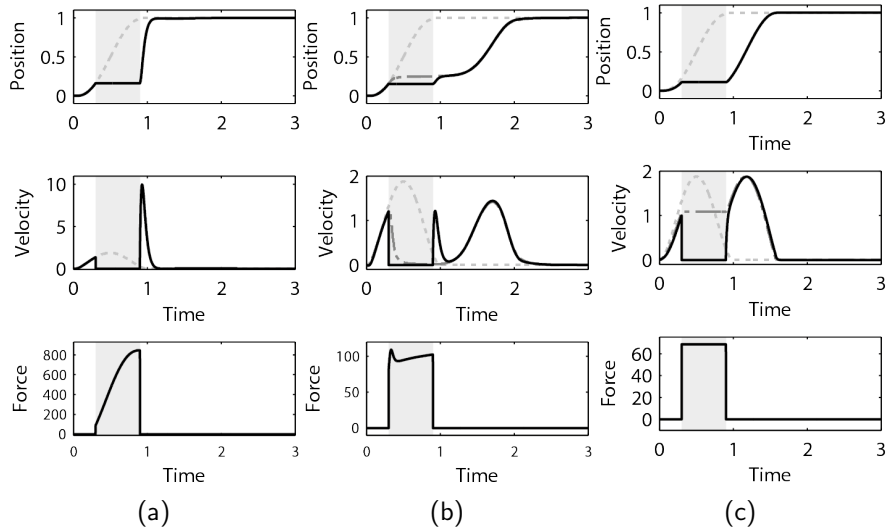


Figure 7.5: Simulation results from setup A,B and C (left to right). The gray area indicates the time that the system was blocked. Nominal positions and velocities are plotted with dashed line. For setup B (middle) the DMP output is plotted with dash-dot line. For setup C (right) the dash-dot line plots the desired velocity, $f(x)$.

$k_C = 10^2$. As can be seen, the behavior in contact is now unpredictable and causing very high contact forces. Another potential pitfall is that if too high k_C is used, position errors occurring without contact could already be enough to slow down the dynamics considerably. As a matter of fact, simply removing the blocking period from the simulation immediately reveals this problem, see Fig. 7.6a. The reason for the coupling terms being active even without the artificial obstacle is clear from Fig. 7.6b, which plots the simulation of setup B without the coupling terms. Because no feedforward acceleration is given in Eq. (7.25), the whole control is driven by feedback errors which also means the coupling becomes active. These problems arise because of the complex cascade of linear and nonlinear systems in (7.26). While it is possible to achieve good performance using these coupling ideas, it must be concluded that tuning the necessary parameters is non-intuitive and may cause hazardous behavior if not set correctly. With PICDS, there is no need to modify the controls. The standard behavior in unexpected contact is already predictable and safe (provided that $\lambda_1 f(x)$ is not a very high value, of course). In this one dimensional simulation we have not used the virtual storage mechanism. Note however, that the only effect it can have on the behavior in contact is to make it safer, since the desired velocity will gradually be slowed down toward zero as the storage becomes depleted.

7.5 Robot Experiment

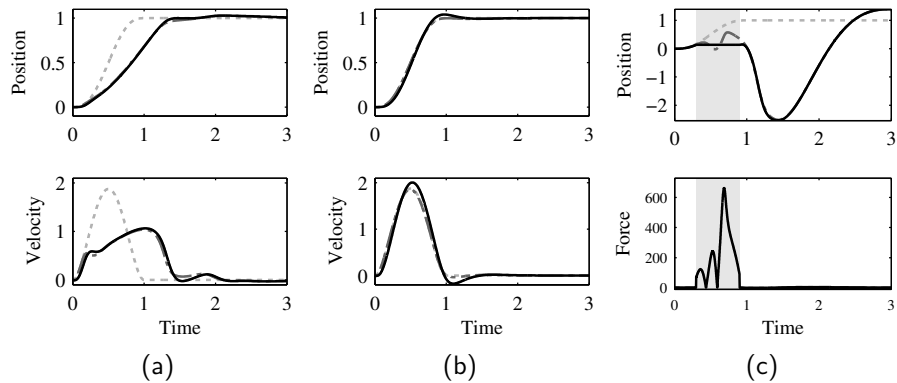


Figure 7.6: **Left:** Simulation of DMP with $k_T = 10^3$, $k_C = 10^4$ but with virtual obstacle removed. The position error occurring due to missing feedforward control is sufficient to activate the coupling and destroy the following of the original trajectory. **Middle:** Tracking performance with DMP without coupling ($k_T = k_C = 0$). **Right:** Simulation with virtual obstacle present. Coupling gains were set to $k_T = 10^4$, $k_C = 10^3$.

In the experiment in Section 6.4.4, we showed how reshaping the dynamics could improve the task model for putting plates into a dish rack. An open-loop integrated reference trajectory from the starting point was used in conjunction with an impedance controller to control the robot during task execution. This controller was hence not passive, and in addition can cause high contact forces in the event of unexpected contact with objects in the environment as shown in Section 7.4. Here, we revisit the same task in a comparative study between the open-loop approach and the passive controller described Section 7.3.3. The experiments in this section use the task model and the same task set up as in Section 6.4.4. The experiments are this time conducted on the KUKA LWR 4+ arm in lieu of the Barrett WAM used in Section 6.4.4. We do not decouple the dynamics nor use nominal control. Instead we investigate how the control methods compare when only a gravity model of the robot is available.

EXPERIMENTAL SETUP

In Section 6.4.4 the attractor of the task DS was placed correctly, so that as the robot reached the attractor, the plate slid into a slot with negligible contact force. In real scenarios, mismatch between environment state and the expected state is unavoidable. To account for this, we conducted three sets of task executions, in each of which the target location of the task DS was offset in different locations behind the real location of the dish rack, see Fig. 7.7. In each set of experiments, 5 task executions were carried out using two different controllers described below.

- A.** The controller from Section 7.3.3 with $\bar{s} = s(0) = 10$, $\lambda_1 = 20$ and $\lambda_2 =$

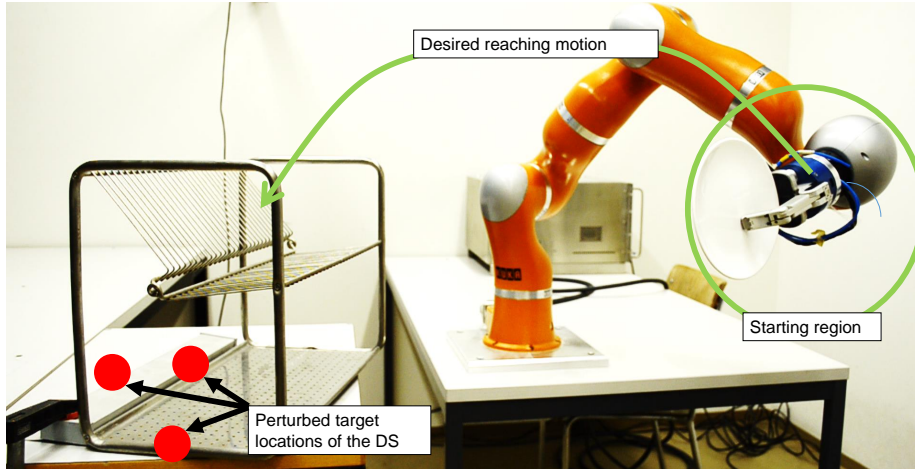


Figure 7.7: The figure shows the experiment setup for the plate insertion task. The motion starts somewhere in the encircled region. The DS is describing a parabolic motion that has an attractor in one of three different perturbed target locations whose approximate position is shown with red dots.

$\lambda_2 = 200$ is used. The value of λ_1 was chosen to the minimal value capable of overcoming static joint friction at the point of departure.

- B.** Openloop trajectory integrated from the initial position of the robot in combination with the impedance controller without inertia shaping described in Section 2.2.4. The stiffness was set to $\mathbf{K} = k\mathbf{I}_{3 \times 3}$ with $k = 100$, the minimum value capable of reaching the final point of the task in free motion.

All task executions were started somewhere in a small region shown in Fig. 7.7. Rotational motion of the end-effector was in both cases simply damped by a high amount (4 Ns/rad) which effectively kept the orientation constant during the task execution.

During the task executions, the Cartesian pose and an estimate of the external force was recorded. The estimate of the external force was computed internally in the KUKA controller using the joint torque sensors of the LWR4+ arm. This estimate is available along with position measurements of the joints via the Fast Research Interface(FRI)(Schreiber et al., 2010). A kinematic model was continuously updated in our C++ software implementation of the two controllers, allowing to convert the Cartesian control force to joint torques using Jacobian transpose control (refer to Section 2.2.4). The FRI was also used to command torques to the robot at a frequency of 1 kHz.

RESULTS

Since very low gains were used, and no inverse dynamics control was applied, it is not expected that either controller would be able to track the nominal

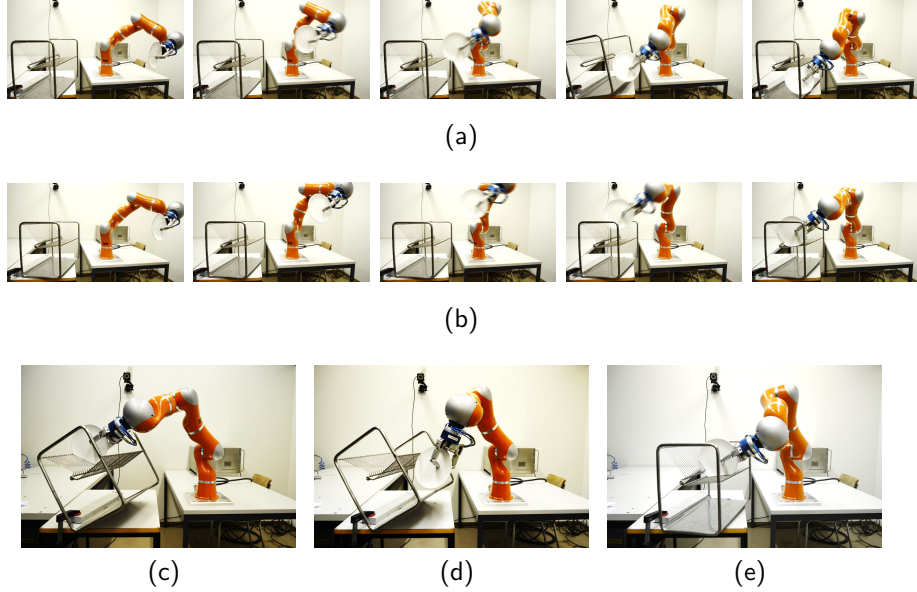


Figure 7.8: **a)** Typical trial with the setup A. The shortcut effect here leads to the plate being stuck under the dish rack. **b)** Typical trial with setup B. The robot follows more accurately the desired motion and yields a low contact force on impact. **c-d)** Examples of final configuration with setup A. **e)** Final configuration with setup B.

motion given by **f** with good accuracy. This is confirmed in Fig. 7.9 which plots the nominal and actual trajectories for each setup for each perturbed location of the dish rack. Especially the ‘shortcut’ tendency of the impedance controller, which is also consistently obvious in Fig. 7.10, which plots the estimated contact force during the trials. Setup B consistently impacts the dish rack before setup A, due to the shortcut effect. PICDS has a clear advantage in terms of respecting shape of the desired reaching motion.

In this particular task, the shortcut effect meant that the robot was approaching the rack from the wrong direction, which sometimes lead to interesting final configurations as depicted in Figures 7.8a, 7.8c and 7.8d. In each of the three perturbed scenarios, simulation A consistently placed the plate correctly because the pattern of approach was respected, see Figures 7.8b and 7.8e.

As is clear from Fig. 7.10, setup A also has an advantage over setup B in terms of contact force after impact. At the time of impact, the reference point for setup B has already reached its final point, which is why there is no gradual ramp-up of the contact force as would normally be expected in contact with a timed trajectory. In the second perturbed location (middle plots in Figures 7.9 and 7.10) setup B resulted in some of the trials landing in a final configuration on the rack and some of the trials landed in a configuration under the rack. This is visible in the divergence of the trajectories near the end-point in Fig. 7.9b right, and also the high variance in the final contact force in Fig. 7.10b right. It should be emphasized that both controllers have been chosen to be as

compliant as possible for this task, but the low stiffness is not enough to ensure a low contact force for positioning errors of this magnitude.

7.6 Discussion and conclusion

We have proposed a novel control system for torque-controlled manipulators for tasks specified by first order dynamical system. We have exemplified it on a particular class of DS, namely LMDS proposed in this thesis, but its applicability extends more generally to autonomous DS models such as SEDS (Khansari-Zadeh and Billard, 2011) and ELM (Lemme et al., 2014).

The proposed approach is related to Passive Velocity Field Control (PVFC) (Li and Horowitz, 1999), which to the best of our knowledge were the first to use velocity fields to achieve the natural passivity relationship (external effort - velocity) that allows to conclude passive interaction with any passive environment as per the passivity theorem. Their approach is based on power-continuous controls achieved via skew-symmetric feedback matrices that depend on the robot dynamics and the desired velocity field. Driving the robot forward along the desired dynamics is achieved by a coupling control with a virtual fly wheel which similarly to our virtual storage provides flexibility by allowing to temporarily store energy in the system. The virtual flywheel is assigned dynamics and a virtual desired velocity field, whose effect on the system performance is non-intuitive. More importantly, the PVFC formulation does not allow to specify the mechanical impedance in an intuitive manner.

Also strongly related is the work of Duindam and Stramigioli presented in (Duindam and Stramigioli, 2003, 2004; Duindam et al., 2004), which much like (Li and Horowitz, 1999) is extensively based on power-continuous controls, but differs from the latter significantly in the way that the robot is driven forward along the desired integral curves. The authors make a somewhat limiting assumption and by considering a single desired, time-independent curve in the work-space, and then use a potential function with a minimum along the desired curve as driving control. Similarly to the PVFC framework, there is no specification of interactive dynamics and the method seems mostly well-conceived for non-contact situations, although limited exchange of energy is ensured if contact should occur. This is possible due to the very clever *power continuous* inverse dynamics mechanism in the form of a skew-symmetric velocity feedback term in (Duindam et al., 2004). We also demonstrated that the nominal control command derived in (Duindam et al., 2004) can be added to PICDS to improve tracking performance in free motion.

In contrast to both (Duindam et al., 2004) and (Li and Horowitz, 1999) which both actively redirect any energy in the system along the desired vector field, our approach uses dissipation as an important part of the control structure. Note that a completely power-continuous controller will keep any energy provided

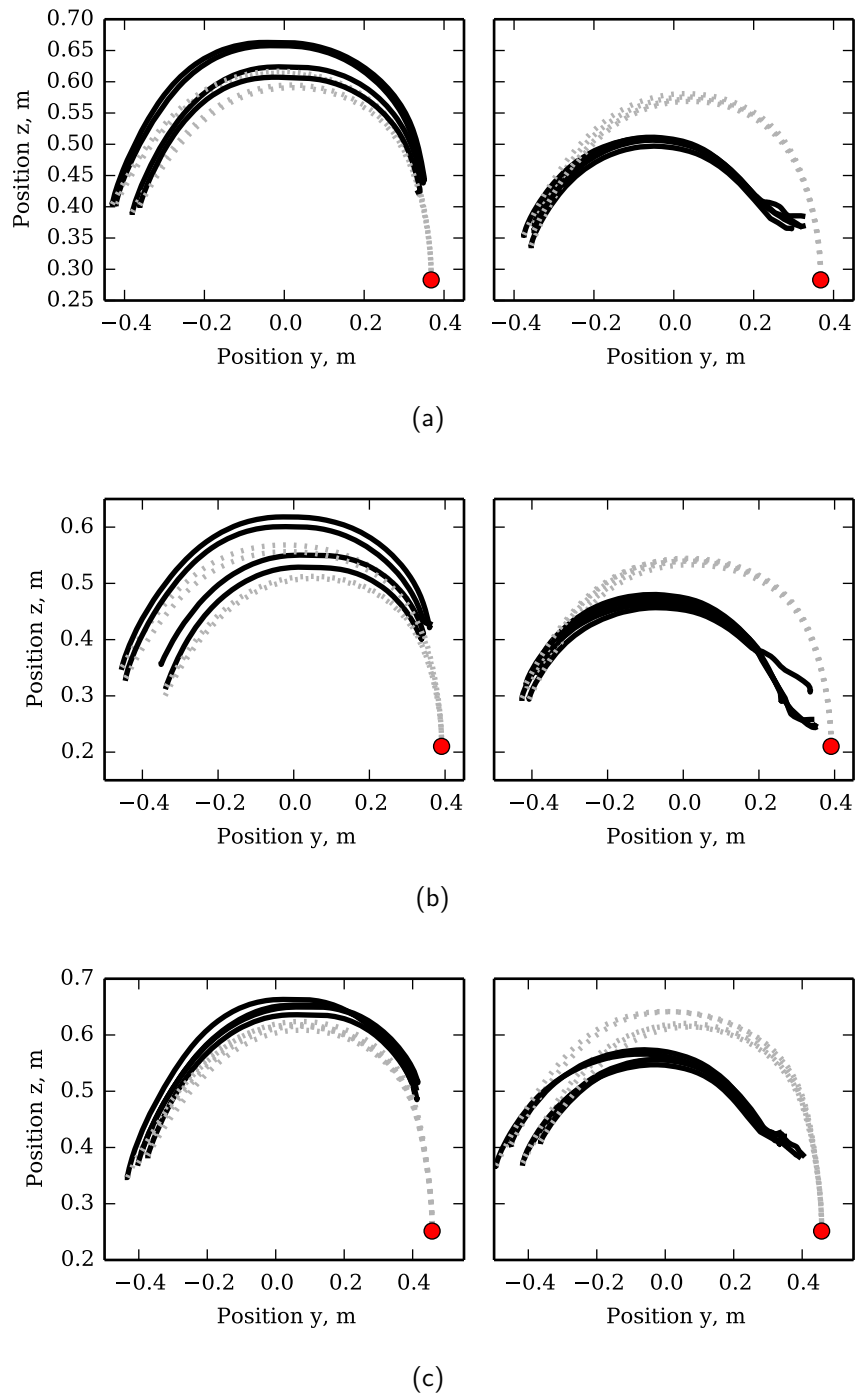
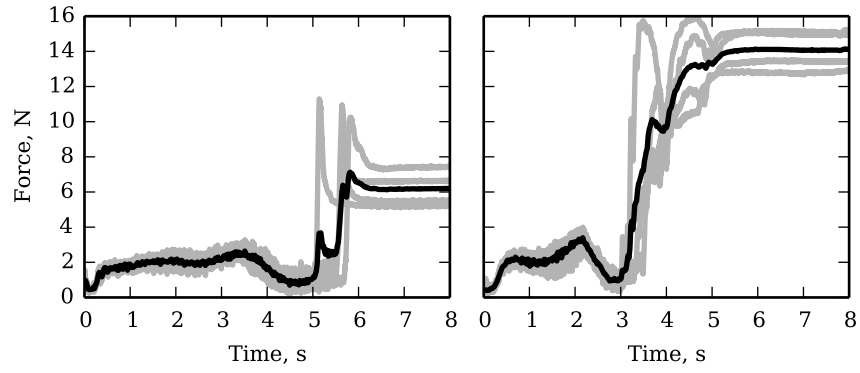
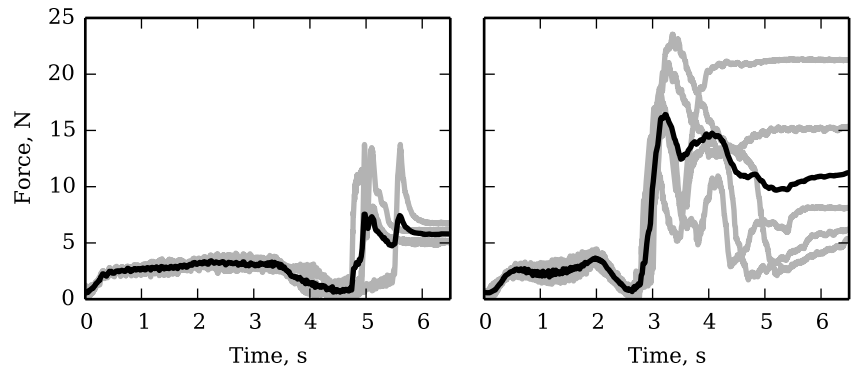


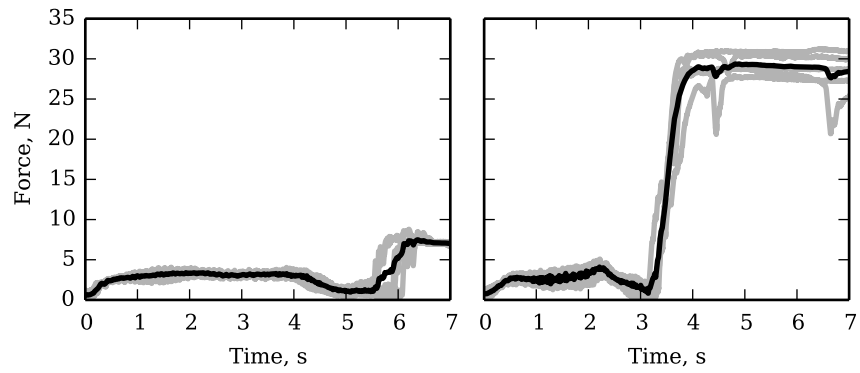
Figure 7.9: Actual (solid lines) and nominal (dotted lines) trajectories for the dish rack experiment. The motion in the YZ-plane is shown since the position in X direction remains almost constant during the motion. Each row shows data from the three different perturbed locations of the dish rack. **Left column:** Setup A. **Right column:** Setup B.



(a)



(b)



(c)

Figure 7.10: The plots show the norm of the estimated external force over time. The raw data are plotted in gray and a temporal average is plotted in black. Each row shows data from the three different perturbed locations of the dish rack. **Left column:** Setup A. **Right column:** Setup B.

too it by the environment, hence the robot may accelerate sharply and change its direction of movement drastically if an strong external force is applied to it. Dissipation is a simpler, safer and more intuitive manner of dealing with undesired energy, albeit more wasteful.

In (Kishi et al., 2003), the non-intuitiveness of tuning the mechanical impedance with the approaches proposed in (Li and Horowitz, 1999) and (Duindam et al., 2004) was already pointed out. An alternative formulation based on standard impedance control was introduced. The speed along the desired curve was then regulated so as to achieve passivity. Their approach offers the possibility to passive tracking in contact, but lacks the flexibility of our approach since a single trajectory is used as reference.

The virtual storage concept used in this chapter is not a novel idea but has been used by several works before us (Stramigioli et al., 2005; Franken et al., 2011). The approach to varying stiffness proposed (Ferraguti et al., 2013) (discussed in detail in Chapter 4) also uses the same general idea of introducing a virtual state variable for implementing temporary and bounded energy storage. Any method based on the concept of a limited virtual energy storage is susceptible to perturbations and may not be able to control the system as desired, depending on external perturbations that are unknown prior to task execution. This is also a potential problem of the PICDS controller proposed in this chapter, which may be forced to fall-back to following the conservative part of the desired dynamics if the virtual storage is depleted. While this may be better than stopping completely, it will generally be detrimental to task performance and the starting level of the virtual storage hence needs to be chosen according to the task, as discussed in Section 7.3.5.

CONCLUSIONS

In this final chapter of the thesis I summarize the main contributions of this thesis. The most important limitations and identified directions for future research are also summarized here.

8.1 Contributions

It is clear that LfD would benefit from having an interface that allows the user to teach the robot impedance information. We are not the only ones to have identified this need (a survey of related works is given in Section 2.1.4), but the approach in Chapter 3 is the first to have provided an interface that allow to teach stiffness variations specifically rather than inferring it implicitly.

The work on stability analysis of variable impedance systems in Chapter 4 was a need that was identified during the development of the methods for teaching and learning varying stiffness in Chapter 3. It was clear that the field had embraced the possibility to improve task performance by varying performance but it had done so without much consideration to the stability implications of varying impedance. The stability conditions derived in Chapter 4 can help to verify that a varying stiffness and damping profile, no matter how it is obtained (e.g. by learning or optimization) will not lead to unstable behavior and as such represent an important first step toward more generally applicable variants of these stability conditions.

Chapters 3 and 4 were focused on impedance control with time-dependent reference trajectories¹. Chapter 5 expanded on this via the use of online reference adaptation based on sensed wrench in a task-specific manner learned from demonstration. More flexibility and intelligent behavior can be encoded if instead an autonomous DS is used to represent the nominal motion. Stability-constrained learning of autonomous DS models is rather complicated and ill-suited for incremental learning. The Locally Modulated Dynamical Systems (LMDS) framework introduced in Chapter 6 allows to *reshape* attractor landscapes with an arbitrary local learning algorithm. Previous DS representations use a parameterized Lyapunov function to ensure stability. The key novelty

¹Although a DS is used in the pouring experiment of Section 3.3.3, it is used in open-loop configuration (see Fig. 2.3) in Chapter 2) and without any incremental refinement

here is that instead of Lyapunov-based stability constraints, we ensure boundedness and non-introduction of spurious attractor points in the choice of DS representation.

The final contribution of this thesis is the development of Passive Interaction Control with Dynamical Systems (PICDS) of Chapter 6. This is a first step toward maximally exploiting the modeling potential of DS models and at the same time leverage on the advantages of impedance control. It should be noted that the stability condition of Chapter 4 is valid in the case of free motion or temporary, bounded perturbations. If an external force that interacts dynamically is admitted in the closed-loop dynamics, the stability proof is (similarly to most other tracking stability proofs) invalidated. The use of an autonomous DS model allows more complex behavior to be modeled but arguably more importantly when used with the PICDS controller of Chapter 7 it ensures stable interaction with any passive environment.

8.2 Limitations and future work

TEACHING IMPEDANCE IN CONTACT WITH THE ENVIRONMENT

The teaching interface developed in Chapter 3 is partially based on physically imposing perturbations on the robot to communicate desired stiffness variations. It can be difficult to use this modality for in-contact tasks, since it may not be possible to wiggle the robot freely if it is in contact or close to being in contact. Least squares estimation procedures using measurement of force and position deviations do not have this limitation (Rozo et al., 2013; Lee et al., 2015). However, these works have the limitation that they can only estimate stiffness in the presence of an interaction force and in addition the estimation is done *after the demonstrations*. However, least squares estimators are straight forward to implement recursively. It would hence be interesting to explore a combination of the interface we proposed in Chapter 3 with a least squares estimation for stiffness estimation when there is a non-zero contact force.

ENSURING STABILITY WITH VARYING STIFFNESS, DAMPING AND INERTIA

The stability conditions of Chapter 4 are restricted in that they require the inertial effects of the manipulator to be compensated. This is the case for the impedance shaping impedance controllers but only if the desired inertia is constant. The method is not applicable to the simpler and more commonly used impedance controllers without inertia shaping. Generalization of the conditions to the case of varying inertia is an interesting challenge for future research.

BETTER UNDERSTANDING OF HUMAN MANIPULATION SKILL IN SEARCH AND INSERTION TASKS

The performance of the peg-in-hole controller proposed in Chapter 5 lags far behind that of a blindfolded human on the same setup. It is important to better understand what capability it is that serves humans so well in this type of tasks, and to transfer these to robots either in a generic or task-specific fashion. A study on human search and insert behavior is at the time of this writing being conducted in the LASA lab.

LMDS IN HIGH DIMENSIONS

The LMDS framework of chapter 6 is based on local full-rank multiplication of an original DS. In particular, we focused on locally scaling and rotating the dynamics. This choice of local transformation makes it straightforward to uniquely determine the parameters for the modulation function corresponding to trajectory training data. This is important since continuous trajectory could otherwise generate a discontinuous parameter evolution which would cause problems for the local regression technique modeling the modulation parameters. In higher dimensions (>3), it is also possible to define rotations but there are several different parameterizations of the rotation possible. A challenge for future work hence, is how to bias the step that translates demonstrated data to modulation parameters so that discontinuous parameter evolutions are avoided.

LEARNING CYCLIC TASKS WITH LMDS

Chapter 6 was focused mainly on modeling discrete motion tasks, i.e. those that have a point where the motion ends. It was shown that cyclic tasks could be encoded if explicitly demonstrated, but it is only possible to reliably achieve a limit cycle in a 2d task representation. Generally, encoding cyclic tasks require to take this fact into account in the choice of representation. For example, in DMP, cyclic motion is achieved by oscillating the phase variable that drives the non-linearities of the system. In LMDS, we do not have this possibility because we have made the design choice to have a DS that is autonomous in the controlled state. Instead, using original dynamics that are cyclic instead of converging to a single attractor seems like a promising route to explore to allow better representation of Cyclic tasks in LMDS.

PARAMETRIC LMDS

We proposed only one learning algorithm for LMDS — GP-MDS based on GPR. While non-parametric regression models like GPR give a scalable capability to model almost arbitrarily complex motions, computationally they typically do not scale very well. This is especially true for GPR. An alternative approach is to prime the system to a level of allowed complexity by determining the

number of parameters beforehand. GMR is an example of such an approach. But GMR is not local in the sense that is required by LMDS. The modified GMR presented in appendix A.1 could however be used. An interesting future direction of research is to formalize and characterize LMDS with a parametric learning algorithm, be it with the modified version of GMR in appendix A.1 or some other method.

LEARNING TASK-BASED ENERGY STORAGE

A difficulty of using the PICDS controller is to appropriately choose \bar{s} , the maximum allowed level of virtually stored energy. This can have a very big impact on the task performance and should hence be chosen carefully. In Section 7.3.5, we proposed a simple heuristic which may be used to get a rough estimate of an appropriate \bar{s} , and we showed that this method worked well for all our simulations and experiments. However, it is not a universally applicable method and especially with a high amount of perturbations that stops the robot and lets it accelerate can potentially deplete the energy storage, resulting in the controller falling back the conservative part of the desired DS. One might argue that \bar{s} could simply be set to an extremely high value — the passivity holds for any finite \bar{s} . However, the implication of allowing the system to store a high amount of energy is that an equally high amount of energy can be hazardously released to the environment. Hence, it is important to set the value of \bar{s} correctly. An interesting direction to explore is if an appropriate level can be estimated through learning, either with a human assistant in an active learning arrangement or by repeated task trials that monitoring the level of virtual storage and adjusting between trials accordingly.

Appendices

APPENDICES FOR CHAPTER 3

A.1 Gaussian Mixture Regression with Parameterized Generalization

This appendix describes a slightly modified version of Gaussian Mixture Regression, which can limit the generalisation in regions poorly covered by the training data and instead fall back on an a priori defined basic distribution in these regions. Consider a GMM with K components that models the joint probability distribution of the variable $\boldsymbol{\xi} \in \mathbb{R}^M$:

$$p(\boldsymbol{\xi}) = \sum_{k=1}^K \pi^k \mathcal{N}(\boldsymbol{\xi}; \mathbf{m}^k, \mathbf{C}^k) \quad (\text{A.1})$$

where $\pi^k > 0$, $\mathbf{m}^k \in \mathbb{R}^M$, $\mathbf{C}^k \in \mathbb{R}^{M \times M}$ for $k = 1 \dots K$ are the priors, means and covariances of the Gaussians in the GMM. Assume that the modeled variable has an input part and an output part $\boldsymbol{\xi} = [\boldsymbol{\xi}_I, \boldsymbol{\xi}_O]^T$. The distribution of the output conditioned on the input can then be written:

$$p(\boldsymbol{\xi}_O | \boldsymbol{\xi}_I) = \sum_{k=1}^K p(k | \boldsymbol{\xi}_I) \mathcal{N}(\boldsymbol{\xi}_O | \boldsymbol{\xi}_I; \mathbf{m}_{O|I}^k, \mathbf{C}_{O|I}^k) \quad (\text{A.2})$$

where

$$\mathbf{m}_{O|I}^k = \mathbf{m}_O^k + \mathbf{C}_{OI}^k (\mathbf{C}_I^k)^{-1} (\boldsymbol{\xi}_I - \mathbf{m}_I^k) \quad (\text{A.3a})$$

$$\mathbf{C}_{O|I}^k = \mathbf{C}_O^k - \mathbf{C}_{OI}^k (\mathbf{C}_I^k)^{-1} \mathbf{C}_{IO}^k \quad (\text{A.3b})$$

with

$$\mathbf{m}^k = \begin{bmatrix} \mathbf{m}_I^k \\ \mathbf{m}_O^k \end{bmatrix}, \quad \mathbf{C}^k = \begin{bmatrix} \mathbf{C}_I^k & \mathbf{C}_{IO}^k \\ \mathbf{C}_{OI}^k & \mathbf{C}_O^k \end{bmatrix} \quad (\text{A.3c})$$

The weighting function $p(k | \boldsymbol{\xi}_I)$ represents the probability of $\boldsymbol{\xi}$ being generated by component k . In standard GMR, this weighting function is defined as:

$$p(k | \boldsymbol{\xi}_I) = \frac{\pi^k \mathcal{N}(\boldsymbol{\xi}_I; \mathbf{m}_I^k, \mathbf{C}_I^k)}{\sum_{i=1}^K \pi^i \mathcal{N}(\boldsymbol{\xi}_I; \mathbf{m}_I^i, \mathbf{C}_I^i)} \quad (\text{A.4})$$

GMR is a functional relationship between the input $\boldsymbol{\xi}_I$ and the output $\boldsymbol{\xi}_O$, which

is achieved by taking the mean of the conditional distribution in Eq. (A.2):

$$E\{\boldsymbol{\xi}_O|\boldsymbol{\xi}_I\} = \sum_{k=1}^K p(k|\boldsymbol{\xi}_I)\mathbf{m}_{O|I}^k \quad (\text{A.5})$$

GMR is a convex sum of linear models, weighted non-linearly across the input space. The normalization of the mixing weights in Eq. (A.4) means that at least one of the linear models will always have a significant contribution to the regression signal. The practical effect of this normalization is generalization, i.e. that the relationships found locally in training data are used in inferring outputs for inputs which lie far from the training data in input space. This is often a desirable effect, since it can reduce the amount of training data needed for good inference. However, sometimes it is inappropriate to generalize in this manner. We propose a novel regression strategy for GMM, which preserves the advantages of the standard GMM/GMR formulation while it allows to control the degree of generalization. We introduce a basic distribution, which describes the default relationship between the input and the output, $p_b(\boldsymbol{\xi}_O|\boldsymbol{\xi}_I)$. We define the global conditional distribution as:

$$p'(\boldsymbol{\xi}_O|\boldsymbol{\xi}_I) = \sum_{k=1}^K p'(k|\boldsymbol{\xi}_I)\mathcal{N}(\boldsymbol{\xi}_O|\boldsymbol{\xi}_I; \mathbf{m}_{O|I}^k, \mathbf{C}_{O|I}^k) \cdots \\ + \left(1 - \sum_{k=1}^K p'(k|\boldsymbol{\xi}_I)\right) p_b(\boldsymbol{\xi}_O|\boldsymbol{\xi}_I) \quad (\text{A.6})$$

We have put prime on the quantities that are different compared to the standard GMR formulation. In contrast to Eq. (A.2), the mixing weights for the components of the GMM does not sum to 1. Instead, $\sum_{k=1}^K p'(k|\boldsymbol{\xi}_I)$ can take any value in the interval $]0, 1[$. The second term describes the contribution from the basic distribution $p_b(\boldsymbol{\xi}_O|\boldsymbol{\xi}_I)$. The key to controlling the generalization are the mixing weights of the components in the GMM, $p'(k|\boldsymbol{\xi}_I)$, which we define as:

$$p'(k|\boldsymbol{\xi}_I) = \frac{\pi^k \mathcal{N}(\boldsymbol{\xi}_I; \mathbf{m}_I^k, \mathbf{C}_I^k)}{\sum_{i=1}^K \pi^i \mathcal{N}(\boldsymbol{\xi}_I; \mathbf{m}_I^i, \mathbf{C}_I^i) + (1 - g(\boldsymbol{\xi}_I))} \quad (\text{A.7a})$$

where

$$g(\boldsymbol{\xi}_I) = \max_k \exp\left(-\frac{1}{2}((\boldsymbol{\xi}_I - \mathbf{m}_I^k)^T (c\mathbf{C}_I^k)^{-1} (\boldsymbol{\xi}_I - \mathbf{m}_I^k))\right) \quad (\text{A.7b})$$

where $c > 0$ is a constant. Compared to the mixing weights in standard GMR in Eq. (A.4), our formulation adds a second term in the denominator. This term describes how distant $\boldsymbol{\xi}_I$ is to the closest Gaussian in the GMM. Note that the distance to each Gaussian is computed with the inverse covariance of the Gaussian as metric. For this computation, the covariance is scaled by $c > 0$. The added term in the denominator will cause the mixing weights for the Gaussians in the GMM to go to zero as the input moves far away from

the Gaussians, while the weight for the default distribution tends to one. The amount of generalisation is controlled with the parameter c , and as can be seen from Eq. (A.7), $c \rightarrow \infty$ leads to full generalization (as in the standard formulation of GMR) while $c \rightarrow 0$ gives weights that drop quickly as the query input moves away from the Gaussians. To move between these two extremes, one can for example control c via a map $\phi : [0, 1[\mapsto [0, \infty[$:

$$c = c(z) = \exp\left(\frac{z}{1-z}\right) - 1 \quad (\text{A.8})$$

where moving z in $[0, 1[$ corresponds to going from minimal to full generalization. To perform regression using this model, we proceed as in standard GMR and take the mean of the distribution in Eq. (A.6), which yields:

$$E\{\boldsymbol{\xi}_O | \boldsymbol{\xi}_I\} = \sum_{k=1}^K p'(k | \boldsymbol{\xi}_I) \mathbf{m}_{I|O}^k + \left(1 - \sum_{k=1}^K p'(k | \boldsymbol{\xi}_I)\right) \mathbf{m}_b \quad (\text{A.9})$$

where \mathbf{m}_b is the mean of the basic distribution $p_b(\boldsymbol{\xi}_O | \boldsymbol{\xi}_I)$. The basic distribution p_b is a key element of this regression technique. It should represent the basic relationship between the input and the output variable. The conceptual difference to standard GMR is that in regions poorly covered in the demonstrations, and hence in the GMM, we infer from the basic distribution instead of insisting that the GMM should generalize globally. Note that for the regression it is sufficient to define the mean of the basic distribution.

A.2 System Usability Scale

The System Usability Scale (SUS) is a method for evaluating the usability of a system. It was introduced in (Brooke, 1996) and has become a widespread method for evaluating user interfaces, including Human-Robot Interaction applications. It consists of a series of the 10 statements listed in Table A.1. For each statement, the user rates his level of agreement on a 5 point Likert scale where 1 corresponds to 'strongly disagree' and 5 corresponds to 'strongly agree'.

Each items contribution to the SUS score ranges from 0 to 4. Each odd numbered statement contributes with the level of agreement minus 1. Each even numbered statement contributes with 5 minus the level of agreement. The SUS score is then obtained by summing the contributions for all items and multiplying by 2.5, yielding a score range of 0 to 100.

A.3 NASA Task Load Index

The NASA Task Load Index (TLX) is a widely used method for evaluating the workload of a task (Hart and Staveland, 1988). It consists of two parts, one of which is aimed at determining how the users experienced the workload, and

Table A.1: The SUS statements

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Table A.2: The TLX subscales for workload evaluation.

<i>Label</i>	<i>Description</i>	<i>Min. Rating</i>	<i>Max. Rating</i>
Mental Demand	How mentally demanding was the task?	Very Low	Very High
Physical Demand	How physically demanding was the task?	Very Low	Very High
Temporal Demand	How hurried or rushed was the pace of the task?	Very Low	Very High
Performance	How successful were you in accomplishing what you were asked to do?	Perfect	Failure
Effort	How hard did you have to work to accomplish your level of performance?	Very Low	Very High
Frustration	How insecure, discouraged, irritated, stressed, and annoyed were you?	Very Low	Very High

a second part which is aimed at defining an individual weighting scheme for combining the answers from the first part to a total workload score.

The first part consists of rating the workload of the task according to six subscales given in Table A.2. The users rate each subscale on a 20 point Likert scale with minimum and maximum value corresponding the phrases listed in Table A.2.

The second part consists of a pairwise presentation of the labels from Table A.2, and for each pair the user is asked to select which one is more important to the workload. For example, if the user is presented with 'mental' and 'physical' she thinks that 'physical' is more important to the workload, she would select the physical label. The number of times each label is chosen is then used to compute a weighted sum of the label scores from the first part, resulting in a workload score ranging from 0 to 100.

PROOF OF THEOREM 4.1

Consider the following Lyapunov candidate function:

$$V_2(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}, t) = \frac{(\dot{\tilde{\mathbf{q}}} + \alpha\tilde{\mathbf{q}})^T \mathbf{H}(\dot{\tilde{\mathbf{q}}} + \alpha\tilde{\mathbf{q}})}{2} + \frac{\tilde{\mathbf{q}}^T \boldsymbol{\beta}(t) \tilde{\mathbf{q}}}{2} \quad (\text{B.1})$$

where $\boldsymbol{\beta}(t)$ is a symmetric, positive definite and continuously differentiable function. Differentiating yields:

$$\dot{V}_2(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}, t) = (\dot{\tilde{\mathbf{q}}} + \alpha\tilde{\mathbf{q}})^T \mathbf{H}(\ddot{\tilde{\mathbf{q}}} + \alpha\dot{\tilde{\mathbf{q}}}) + \tilde{\mathbf{q}}^T \dot{\boldsymbol{\beta}}(t) \dot{\tilde{\mathbf{q}}} + \frac{1}{2} \tilde{\mathbf{q}}^T \dot{\boldsymbol{\beta}}(t) \tilde{\mathbf{q}} \quad (\text{B.2})$$

Substituting the closed loop dynamics from Eq. (2.18) and rearranging yields:

$$\begin{aligned} \dot{V}_2(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}, t) &= \dot{\tilde{\mathbf{q}}}^T \{ \alpha \mathbf{H} - \mathbf{D}(t) \} \dot{\tilde{\mathbf{q}}} \\ &\quad + \dot{\tilde{\mathbf{q}}}^T \{ \beta(t) + \alpha^2 \mathbf{H} - \mathbf{K}(t) - \alpha \mathbf{D}(t) \} \tilde{\mathbf{q}} \\ &\quad + \tilde{\mathbf{q}}^T \left\{ \frac{1}{2} \dot{\boldsymbol{\beta}}(t) - \alpha \mathbf{K}(t) \right\} \tilde{\mathbf{q}} \end{aligned} \quad (\text{B.3})$$

In order to eliminate the cross-term between $\tilde{\mathbf{q}}$ and $\dot{\tilde{\mathbf{q}}}$, define $\beta(t)$ as:

$$\beta(t) = \mathbf{K}(t) + \alpha \mathbf{D}(t) - \alpha^2 \mathbf{H} \quad \Rightarrow \quad \dot{\beta}(t) = \dot{\mathbf{K}}(t) + \alpha \dot{\mathbf{D}}(t) \quad (\text{B.4})$$

Substituting $\beta(t)$ and $\dot{\beta}(t)$ into Eq. (B.3) then yields:

$$\dot{V}_2(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}, t) = \dot{\tilde{\mathbf{q}}}^T \{ \alpha \mathbf{H} - \mathbf{D}(t) \} \dot{\tilde{\mathbf{q}}} + \tilde{\mathbf{q}}^T \left\{ \frac{1}{2} \dot{\mathbf{K}}(t) + \frac{\alpha}{2} \dot{\mathbf{D}}(t) - \alpha \mathbf{K}(t) \right\} \tilde{\mathbf{q}} \quad (\text{B.5})$$

Note that $\beta(t)$ is positive definite (Substitute condition 1 from Theorem 4.1 in Eq. (B.4)), which implies that V_2 is positive definite. V_2 is also decrescent, since it is dominated e.g. by: $V_2 < (\dot{\tilde{\mathbf{q}}} + \alpha\tilde{\mathbf{q}})^T \mathbf{H}(\dot{\tilde{\mathbf{q}}} + \alpha\tilde{\mathbf{q}}) + \|\tilde{\mathbf{q}}\|^2 \max_t \bar{\lambda}(\boldsymbol{\beta}(t))$. Furthermore, V_2 is radially unbounded, and substituting condition 2 from Theorem 1 in Eq. (B.5) confirms $V_2 < 0$ for all $\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}} \neq 0$, which concludes the proof.

APPENDICES FOR CHAPTER 7

C.1 Smooth coupling functions

In Section 7.3.3, the scalar functions α, β_s and β_R were introduced. In this appendix, we describe how these functions were chosen in this thesis. We restate the specifications of α, β_s and β_R from Section 7.3.3:

$$\begin{cases} 0 \leq \alpha(s) \leq 1 & s < \bar{s} \\ \alpha(s) = 0 & s \geq \bar{s} \end{cases} \quad (\text{C.1a})$$

$$\begin{cases} \beta_s(s, z) = 0 & s \leq 0 \text{ and } z \geq 0 \\ \beta_s(s, z) = 0 & s \geq \bar{s} \text{ and } z \leq 0 \\ 0 \leq \beta(s, z) \leq 1 & \text{elsewhere} \end{cases} \quad (\text{C.1b})$$

$$\begin{cases} \beta_R(s, z) = \beta_s(s, z) & z \geq 0 \\ \beta_R(s, z) \geq \beta_s(s, z) & z < 0 \end{cases} \quad (\text{C.1c})$$

These criteria allow a considerable amount of freedom. Note that it is not necessary that the functions be continuous in z or s . However, to avoid sharp changes in control if the storage is depleted, we will make use of smooth step functions for defining these coupling functions. To this end, we introduce the functions $\mathcal{H}_{a,b}$ and $\mathcal{H}_{a,b}^-$ denoting smooth step functions defined as:

$$\begin{cases} \mathcal{H}_{a,b}(x) = 0 & x < a \\ \mathcal{H}_{a,b}(x) = 6\left(\frac{x-a}{b-a}\right)^5 - 15\left(\frac{x-a}{b-a}\right)^4 + 10\left(\frac{x-a}{b-a}\right)^3 & a \leq x \leq b \\ \mathcal{H}_{a,b}(x) = 1 & x > b \end{cases} \quad (\text{C.2a})$$

$$\begin{cases} \mathcal{H}_{a,b}^-(x) = 1 - \mathcal{H}_{a,b}(x) \end{cases} \quad (\text{C.2b})$$

The fifth order polynomial has the role of transitioning smoothly from 0 to 1, while matching first and second order derivatives (=0) at $x = a$ and $x = b$. Fig. C.1 illustrates these functions.

Starting by $\alpha(s)$, this function has the role of limiting the harvest of damping work which would otherwise be dissipated. This harvest (first term in Eq. (7.7)), is only allowed as long as there is still space in the virtual storage, i.e. $s < \bar{s}$.

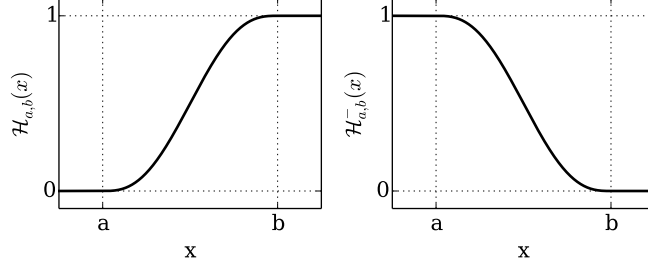


Figure C.1: The figure shows the smooth step functions used to compose α , β_s and β_R .

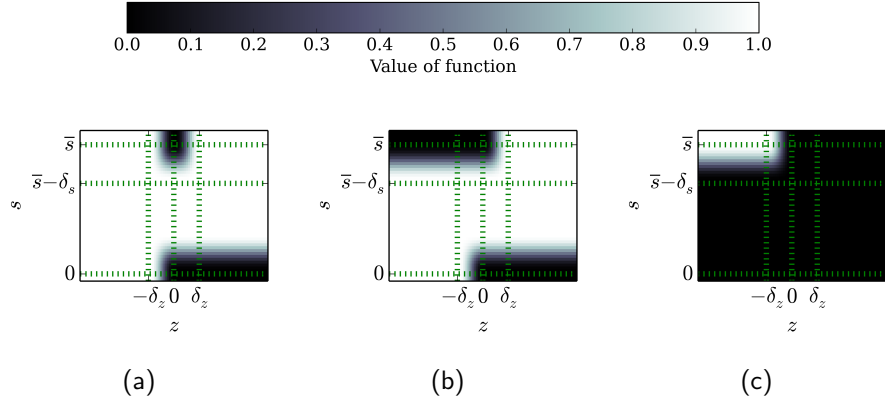


Figure C.2: The figure shows the β -functions and their difference over their 2d domain. Relatively high values of δ_s, δ_z are used for illustrations purposes. **Left:** β_r **Middle:** β_s **Right:** $\beta_r - \beta_s$

We define α simply as an inverted smooth step function:

$$\alpha(s) = \mathcal{H}_{\bar{s}-\delta_s, \bar{s}}^-(s) \quad (\text{C.3})$$

where δ_s is a smoothness parameter determining the length of the transition from 1 to 0. In this work, $\delta_s = 0.1\bar{s}$ was always used.

The β_s function is defined as:

$$\beta_s(s, z) = 1 - \mathcal{H}_{-\delta_z, 0}(z) \mathcal{H}_{\bar{s}, \bar{s}+\delta_s}^-(s) - \mathcal{H}_{0, \delta_z}^-(z) \mathcal{H}_{\bar{s}-\delta_s, \bar{s}}(s) \quad (\text{C.4})$$

where $\delta_z > 0$ is a smoothness parameter. The function is plotted in Fig. C.2b.

Finally, the β_R function is defined as:

$$\beta_R(s, z) = \left(1 - \mathcal{H}_{-\delta_z, 0}(z) \mathcal{H}_{\bar{s}, \bar{s}+\delta_s}^-(s)\right) \left(1 - \mathcal{H}_{-\delta_z, 0}(z) \mathcal{H}_{0, \delta_z}^-(z) \mathcal{H}_{\bar{s}-\delta_s, \bar{s}}(s)\right) \quad (\text{C.5})$$

Fig. C.2a plots β_R . To avoid unnecessarily scaling the non-conservative velocity component, δ_z should be chosen small. In this work, we have always used $\delta_z = 0.01$.

Note that the functions specified here are just examples of functions that

meet the criteria in Eq. (C.1). There are of course many other possibilities, but these functions have worked well in this work.

C.2 Proof of Proposition 7.1

The rate of change of W is:

$$\dot{W}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \dot{\boldsymbol{\xi}}^T \mathbf{M}(\boldsymbol{\xi}) \ddot{\boldsymbol{\xi}} + \frac{1}{2} \dot{\boldsymbol{\xi}}^T \dot{\mathbf{M}}(\boldsymbol{\xi}) \dot{\boldsymbol{\xi}} + \lambda_1 \nabla V_f^T \dot{\boldsymbol{\xi}} \quad (\text{C.6})$$

Substituting $\mathbf{M}\ddot{\boldsymbol{\xi}}$ from Eq. (7.1) with $\boldsymbol{\tau}_c$ given by Eq. (7.5) yields:

$$\dot{W}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \frac{1}{2} \dot{\boldsymbol{\xi}}^T (\dot{\mathbf{M}} - 2\mathbf{C}) \dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}^T \mathbf{D} \dot{\boldsymbol{\xi}} + \dot{\boldsymbol{\xi}}^T \boldsymbol{\tau}_e + \lambda_1 \dot{\boldsymbol{\xi}}^T \mathbf{f}(\boldsymbol{\xi}) + \lambda_1 \nabla V_f^T \dot{\boldsymbol{\xi}} \quad (\text{C.7})$$

where dependencies of \mathbf{M}, \mathbf{D} on $\boldsymbol{\xi}$ and \mathbf{C} of $\dot{\boldsymbol{\xi}}$ has been omitted for cleanliness of notation. In Eq. (C.12) the first term is null due to the skew-symmetry of the matrix $\dot{\mathbf{M}} - 2\mathbf{C}$ (refer to Section 2.2.3) and the last to terms cancel because $\mathbf{f}(\boldsymbol{\xi}) = -\nabla V_f$. Hence, we have:

$$\dot{W}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = -\dot{\boldsymbol{\xi}}^T \mathbf{D} \dot{\boldsymbol{\xi}} + \dot{\boldsymbol{\xi}}^T \boldsymbol{\tau}_e \quad (\text{C.8})$$

which proves passivity according to Def. 2.7.

C.3 Proof of Theorem 7.1

First, note that $0 < s(0) \leq \bar{s} \Rightarrow 0 \leq s(t) \leq \bar{s}, \forall t > t_0$, trivially from Equations (7.7), (7.8) and (7.9). Consider the storage function $W(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}, s) = \frac{1}{2} \dot{\boldsymbol{\xi}}^T \mathbf{M} \dot{\boldsymbol{\xi}} + \lambda_1 V_c(\boldsymbol{\xi}) + s$, where $V_c(\boldsymbol{\xi})$ is the potential function associated with $\mathbf{f}_c(\boldsymbol{\xi})$. The rate of change of W is:

$$\dot{W}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \dot{\boldsymbol{\xi}}^T \mathbf{M} \ddot{\boldsymbol{\xi}} + \frac{1}{2} \dot{\boldsymbol{\xi}}^T \dot{\mathbf{M}} \dot{\boldsymbol{\xi}} + \lambda_1 \nabla V_c^T \dot{\boldsymbol{\xi}} + \dot{s} \quad (\text{C.9})$$

Substituting $\mathbf{M}\ddot{\boldsymbol{\xi}}$ from Eq. (7.1) with $\boldsymbol{\tau}_c$ given by Eq. (7.10) and using the skew-symmetry of $\dot{\mathbf{M}} - 2\mathbf{C}$ as in Proposition 7.1 yields:

$$\dot{W}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = -\dot{\boldsymbol{\xi}}^T \mathbf{D} \dot{\boldsymbol{\xi}} + \dot{\boldsymbol{\xi}}^T \boldsymbol{\tau}_e + \beta_R(z, s) \lambda_1 z + \lambda_1 \dot{\boldsymbol{\xi}}^T \mathbf{f}_c(\boldsymbol{\xi}) + \lambda_1 \nabla V_c^T \dot{\boldsymbol{\xi}} + \dot{s} \quad (\text{C.10})$$

The second-to-last two terms cancel because $\mathbf{f}_c(\boldsymbol{\xi}) = -\nabla V_c(\boldsymbol{\xi})$. Substituting \dot{s} from Eq. (7.7) then yields:

$$\dot{W}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = -\underbrace{\{1 - \alpha(s)\}}_{\geq 0} \dot{\boldsymbol{\xi}}^T \mathbf{D} \dot{\boldsymbol{\xi}} + \zeta(z, s) \lambda_1 z + \dot{\boldsymbol{\xi}}^T \boldsymbol{\tau}_e \quad (\text{C.11})$$

where $\zeta(z, s) = \beta_R(z, s) - \beta_s(z, s)$ has been introduced to ease the notation. Note that by Eq. (7.8) we have $1 - \alpha(s) \geq 0$. By Equations (7.9) and (7.11) we

have that $\zeta(z, s) = 0$ for all $z > 0$ and $\zeta(z, s) \geq 0$ for $z < 0$. Hence, we have:

$$\dot{W}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) \leq \dot{\boldsymbol{\xi}}^T \boldsymbol{\tau}_e \quad (\text{C.12})$$

which concludes the proof.

REFERENCES

- Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. *Proceedings of the 21st International Conference on Machine Learning (ICML)*, pages 1–8, 2004. doi: 10.1145/1015330.1015430. 2.1.4
- Fares J. Abu-Dakka, Bojan Nemec, Aljaž Kramberger, Anders Glent Buch, Norbert Krüger, and Ales Ude. Solving peg-in-hole tasks by human demonstration and exception strategies. *Industrial Robot: An International Journal*, 41(6): 575–584, 2014. ISSN 0143-991X. doi: 10.1108/IR-07-2014-0363. 2.1.5, 5.7
- A. Ajoudani, N. Tsagarakis, and A. Bicchi. Tele-impedance: Teleoperation with impedance regulation using a body-machine interface. *The International Journal of Robotics Research*, 31(13):1642–1656, October 2012. ISSN 0278-3649. doi: 10.1177/0278364912464668. 2.1.4
- A. Albu-Schaffer, S. Haddadin, Ch Ott, A Stemmer, T. Wimböck, and G Hirzinger. The DLR lightweight robot: design and control concepts for robots in human environments. *Industrial Robot: An International Journal*, 34(5):376–385, 2007a. 2.1.1
- A. Albu-Schaffer, C. Ott, and G. Hirzinger. A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots. *The International Journal of Robotics Research*, 26(1):23–39, January 2007b. ISSN 0278-3649. doi: 10.1177/0278364907073776. 2.1.1
- A. Albu-Schaffer, Oliver Eiberger, Markus Grebenstein, S. Haddadin, Christian Ott, T. Wimbock, Sebastian Wolf, G. Hirzinger, and Thomas Wimb Ock. Soft robotics. *Robotics & Automation Magazine, IEEE*, 15(3):20–30, 2008. 1.2.1, 2.1.1
- C. H. An and J. M. Hollerbach. The Role of Dynamic Models in Cartesian Force Control of Manipulators. *The International Journal of Robotics Research*, 8: 51–72, 1989. ISSN 0278-3649. doi: 10.1177/027836498900800403. 2.1.1
- Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A Survey of Robot Learning from Demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, May 2009. ISSN 09218890. doi: 10.1016/j.robot.2008.10.024. 1.1
- H Asada. Teaching and learning of compliance using neural nets: Representation and generation of nonlinear compliance. In *IEEE International Conference on Robotics and Automation*, pages 1237–1244, 1990. 2.1.5
- H. Asada and H. Izumi. Automatic program generation from teaching data for the hybrid control of robots. *IEEE Transactions on Robotics and Automation*, 5(2):166–173, April 1989. ISSN 1042296X. doi: 10.1109/70.88037. 2.1.3, 2.1.4

- Haruhiko Asada. Representation and learning of nonlinear compliance using neural nets. *IEEE Transactions on Robotics and Automation*, 9(6):863–867, 1993. 2.1.4, 2.1.5, 5.1, 5.7
- Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally Weighted Learning. *Artificial Intelligence Review*, 11:11–73, 1997. ISSN 00070912. doi: 10.1093/bja/ael290. 6.5
- F Badano, A Jutard, and M Betemps. Chamferless robotic assembly using random search. In *Fifth International Conference on Advanced Robotics*, 1991. ISBN 0780300785. doi: 10.1109/ICAR.1991.240501. 2.1.5
- Leonardo Balletti, Alessio Rocchi, Felipe Belo, Manuel Catalano, Manolo Garabini, Giorgio Grioli, and Antonio Bicchi. Towards variable impedance assembly: The VSA peg-in-hole. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 402–408. Ieee, November 2012. ISBN 978-1-4673-1369-8. doi: 10.1109/HUMANOIDS.2012.6651551. 2.1.5, 5.7
- Randall D. Beer. A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72(1-2):173–215, 1995. ISSN 00043702. doi: 10.1016/0004-3702(94)00005-L. 1.1, 1.2.2
- Antonio Bicchi, Giovanni Tonietti, Michele Bavaro, and Marco Piccigallo. Variable Stiffness Actuators for Fast and Safe Motion Control. In *The Eleventh International Symposium on Robotics Research*, volume 15, pages 527–536. Springer, 2005. ISBN 978-3-540-23214-8. doi: 10.1007/11008941_56. 2.1.1
- Aude Billard, Sylvain Calinon, R. Dillmann, and S. Schaal. Handbook of Robotics Chapter 59: Robot Programming by Demonstration. In *Handbook of Robotics*. Springer, 2008. ISBN 978-1-4244-2057-5. doi: 10.1109/IROS.2008.4650593. 1.1, 1.2.1, 2.1.3
- David Braun, Matthew Howard, and Sethu Vijayakumar. Optimal variable stiffness control: formulation and application to explosive movement tasks. *Autonomous Robots*, 1(17):237–253, 2012. ISSN 0929-5593. doi: 10.1007/s10514-012-9302-3. 2.1.4
- David J Braun, Florian Petit, Felix Huber, Sami Haddadin, Patrick Van Der Smagt, Alin Albu-Schaffer, and Sethu Vijayakumar. Robots driven by compliant actuators: Optimal control under actuation constraints. *IEEE Transactions on Robotics*, 29(5):1085–1101, 2013. ISSN 15523098. doi: 10.1109/TRO.2013.2271099. 2.1.4
- R W Brockett. Hybrid Models for Motion Control Systems. *Essays on Control: Perspectives in the Theory and its Applications*, page 29, 1993. 2.1.1, 2.1.1
- J Brooke. SUS - A quick and dirty usability scale. *Usability Evaluation in Industry*, 189(194):6–7, 1996. 3.3.4, A.2
- J. Buchli, E. Theodorou, F. Stulp, and S. Schaal. Variable Impedance Control—A Reinforcement Learning Approach. *Robotics: Science and Systems*, 2010. 2.1.4
- J. Buchli, F. Stulp, E. Theodorou, and S. Schaal. Learning variable impedance control. *The International Journal of Robotics Research*, 30(7):820–833, 2011. ISSN 0278-3649. doi: 10.1177/0278364911402527. 2.1.4, 2.1.6, 2.1.8, 4.4.2

- D Bullock and S Grossberg. The Vite Model: A Neural Command Circuit for Generating Arm and Articulator Trajectories. *Dynamic patterns in complex systems*, pages 305–326, 1988. 1.1, 1.2.2, 2.1.6
- E Burdet and M Nuttin. Learning complex tasks using a stepwise approach. *Journal of Intelligent and Robotic Systems*, 24(1):43–68, 1999. 2.1.5
- 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*, 414(6862):446–9, November 2001. ISSN 0028-0836. doi: 10.1038/35106566. 1.1, 1.2.1, 2.1.2, 2.1.4
- F. Caccavale, C. Natale, B. Siciliano, and L. Villani. Six-DOF impedance control based on angle/axis representations. *IEEE Transactions on Robotics and Automation*, 15(2):289–300, April 1999. ISSN 1042296X. doi: 10.1109/70.760350. 2.2.4, 2.2.4
- Sylvain Calinon and Aude Billard. Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 255–262. ACM, 2007. ISBN 9781595936172. 2.1.7
- Sylvain Calinon, Florent Guenter, and Aude Billard. On learning the statistical representation of a task and generalizing it to various contexts. *Proceedings - IEEE International Conference on Robotics and Automation*, 2006(Icra): 2978–2983, 2006. ISSN 10504729. doi: 10.1109/ROBOT.2006.1642154. 2.1.3
- Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 37(2):286–98, April 2007. ISSN 1083-4419. 2.1.3, 2.1.7
- Sylvain Calinon, Irene Sardellitti, and D.G. Caldwell. Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 249–254, 2010. 2.1.4, 3.2.1
- Sylvain Calinon, Zhibin Li, Tohid Alizadeh, Nikos G Tsargarakis, and DG Caldwell. Statistical dynamical systems for skills acquisition in humanoids. *International Conference on Humanoid Robots*, 2012. 2.1.6
- Sylvain Calinon, Petar Kormushev, and Darwin G. Caldwell. Compliant skills acquisition and multi-optima policy search with EM-based reinforcement learning. *Robotics and Autonomous Systems*, 61(4):369–379, 2013. ISSN 09218890. doi: 10.1016/j.robot.2012.09.012. 2.1.4
- Sylvain Calinon, Danilo Bruno, and Darwin G. Caldwell. A task-parameterized probabilistic model with minimal intervention control. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3339–3344, May 2014. doi: 10.1109/ICRA.2014.6907339. 2.1.4, 4.4.2
- Giorgio Cannata, Marco Maggiali, Giorgio Metta, and Giulio Sandini. An embedded artificial skin for humanoid robots. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 434–438, 2008. doi: 10.1109/MFI.2008.4648033. 3.3.3

- Manuel Catalano, Giorgio Grioli, Manolo Garabini, Felipe Weilemann Belo, Andrea Di Basco, Nikos Tsagarakis, and Antonio Bicchi. A Variable Damping module for Variable Impedance Actuation. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2666–2672, 2012. ISBN 9781467314039. doi: 10.1109/ICRA.2012.6224938. 2.1.1
- I Cathers, N O’Dwyer, and P Neilson. Tracking performance with sinusoidal and irregular targets under different conditions of peripheral feedback. *Experimental brain research*, 111(3):437–46, October 1996. ISSN 0014-4819. 3.3.1
- Thomas Cederborg, Ming Li, Adrien Baranes, and PY Oudeyer. Incremental local online Gaussian Mixture Regression for imitation learning of multiple tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 267–274, Taipei, Taiwan, 2010. 2.1.7
- E. Cha, K. Kronander, and A. Billard. Combined Kinesthetic and Simulated Interface for Teaching Robot Motion Models. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2015. 1.4
- S. Chiaverini and L. Sciavicco. The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):361–373, 1993. ISSN 1042296X. doi: 10.1109/70.246048. 2.1.1
- P Corke. *Robotics, Vision & Control*. Springer, 2011. 7.4.1
- M. R. Cutkosky and P. K. Wright. Active Control of a Compliant Wrist in Manufacturing Tasks. *Journal of Engineering for Industry*, 108(1):36, 1986. ISSN 00220817. doi: 10.1115/1.3187038. 2.1.5
- Nathan Delson and Harry West. Robot programming by human demonstration: adaptation and inconsistency in constrained motion. In *International Conference on Robotics and Automation*, pages 30–36, 1996. ISBN 0-7803-2988-0. doi: 10.1109/ROBOT.1996.503569. 2.1.3
- AP Dempster, NM Laird, and DB Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977. 2.1.7, 3.3.3, 5.3
- Samuel Hunt Drake. *Using compliance in lieu of sensory feedback for automatic assembly*. PhD thesis, 1977. 2.1.1, 2.1.5
- Harris Drucker, CJC Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *Advances in neural information processing systems*, (x):155–161, 1997. 6.5
- B. Dufay and J.-C. Latombe. An Approach to Automatic Robot Programming Based on Inductive Learning, 1984. ISSN 0278-3649. 2.1.3
- V. Duindam, S. Stramigioli, and J.M.a. Scherpen. Passive Compensation of Nonlinear Robot Dynamics. *IEEE Transactions on Robotics and Automation*, 20(3):480–487, June 2004. ISSN 1042-296X. doi: 10.1109/TRA.2004.824693. 2.1.8, 7.1, **B.**, 7.6
- Vincent Duindam and Stefano Stramigioli. Passive asymptotic curve tracking. In *2nd IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control*, number 1, pages 229–234, Seville, 2003. 7.6

- Vincent Duindam and Stefano Stramigioli. Port-Based Asymptotic Curve Tracking for Mechanical Systems. *European Journal of Control*, 10(5):411–420, 2004. ISSN 09473580. doi: 10.3166/ejc.10.411-420. 2.1.8, 4.1, 7.6
- Ahmetcan Erdogan and Volkan Patoglu. Slacking prevention during assistive contour following tasks with guaranteed coupled stability. *IEEE International Conference on Intelligent Robots and Systems*, pages 1587–1594, 2012. ISSN 21530858. doi: 10.1109/IROS.2012.6386099. 2.1.8
- Ahmetcan Erdogan, AC Satici, and Volkan Patoglu. Passive Velocity Field Control of a Forearm-Wrist Rehabilitation Robot. In *IEEE International Conference on Rehabilitation Robotics*, volume 2011, January 2011. ISBN 9781424498628. doi: 10.1109/ICORR.2011.5975433. 2.1.8
- Federica Ferraguti, Cristian Secchi, and Cesare Fantuzzi. A tank-based approach to impedance control with variable stiffness. In *2013 IEEE International Conference on Robotics and Automation*, pages 4948–4953. Ieee, May 2013. ISBN 978-1-4673-5643-5. doi: 10.1109/ICRA.2013.6631284. 2.1.2, 4.1, 4.3, 4.2, 4.4.1, 4.4.1, 4.4.1, 4.3, 4.5, 7.3.3, 7.6
- Michel Franken, Stefano Stramigioli, Sarthak Misra, Cristian Secchi, and Alessandro MacChelli. Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency. *IEEE Transactions on Robotics*, 27:741–756, 2011. ISSN 15523098. doi: 10.1109/TRO.2011.2142430. 2.1.2, 7.3.3, 7.6
- David W Franklin, Etienne Burdet, Keng Peng Tee, Rieko Osu, Chee-Meng Chew, Theodore E Milner, and Mitsuo Kawato. CNS learns stable, accurate, and efficient movements using a simple algorithm. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 28(44):11165–73, October 2008. ISSN 1529-2401. doi: 10.1523/JNEUROSCI.3099-08.2008. 1.2.1, 2.1.4
- Andrej Gams, Bojan Nemec, Auke Jan Ijspeert, and Aleš Ude. Coupling movement primitives: Interaction with the environment and bimanual tasks. *IEEE Transactions on Robotics*, 30(4):816–830, 2014. ISSN 15523098. doi: 10.1109/TRO.2014.2304775. 7.4.2
- Gowrishankar Ganesh, Nathanael Jarrass, Sami Haddadin, A. Albu-Schaffer, and Etienne Burdet. A versatile biomimetic controller for contact tooling and haptic exploration. In *IEEE International Conference on Robotics and Automation, ICRA*, pages 3329–3334, 2012. ISBN 9781467314053. 2.1.4
- Manolo Garabini, Andrea Passaglia, Felipe Belo, Paolo Salaris, and Antonio Bicchi. Optimality Principles in Variable Stiffness Control : The VSA Hammer. In *International Conference on Intelligent Robots and Systems (IROS)*, 2011. 2.1.4
- Manolo Garabini, Andrea Passaglia, Felipe Belo, Paolo Salaris, and Antonio Bicchi. Optimality principles in stiffness control: The VSA kick. In *IEEE Intl. Conf. on Robotics and Automation*, number 1, pages 3341–3346, 2012. ISBN 9781467314053. 3.3.6
- H. Gomi and M. Kawato. Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement. *Science (New York, N. Y.)*, 272(5258):117–120, April 1996. ISSN 0036-8075. 2.1.4

- H Gomi and R Osu. Task-dependent viscoelasticity of human multijoint arm and its spatial characteristics for interaction with environments. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 18(21): 8965–8978, 1998. ISSN 0270-6474. 2.1.2, 2.1.4
- E. Gribovskaya, S. M. Khansari-Zadeh, and a. Billard. Learning Non-linear Multivariate Dynamics of Motion in Robotic Manipulators. *The International Journal of Robotics Research*, 30(1):80–117, 2011. ISSN 0278-3649. doi: 10.1177/0278364910376251. 2.1.6
- V. Gullapalli, A.G. Barto, and R.A. Grupen. Learning admittance mappings for force-guided assembly. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2633–2638. IEEE Comput. Soc. Press, 1994. ISBN 0-8186-5330-2. doi: 10.1109/ROBOT.1994.351117. 2.1.5
- Sami Haddadin, Michael Weis, Sebastian Wolf, and A. Albu-Schaffer. Optimal control for maximizing link velocity of robotic variable stiffness joints. *IFAC World Congress*, pages 6863–6871, 2011. 2.1.4
- R Ham, T Sugar, B Vanderborght, KW Hollander, and D Lefeber. Compliant actuator designs. *IEEE Robotics & Automation Magazine*, 16(3)(September): 81–94, 2009. 2.1.1
- Blake Hannaford and Jee Hwan Ryu. Time-domain passivity control of haptic interfaces. *IEEE Transactions on Robotics and Automation*, 18(1):1–10, 2002. ISSN 1042296X. doi: 10.1109/70.988969. 2.1.2
- Blake Hannaford, Laurie Wood, Douglas A. McAfee, and Haya Zak. Performance evaluation of a six-axis generalized force-reflecting teleoperator. *IEEE transactions on systems, man, and cybernetics*, 21(3):620–633, 1991. 2.1.5, 5.1
- SG Hart and LE Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Human mental workload*, pages 139–183, 1988. 3.3.4, A.3
- Neville Hogan. Impedance Control: An Approach to Manipulation: Part II—Implementation. *Journal of Dynamic Systems, Measurement, and Control*, 107:8–16, 1985a. ISSN 00220434. doi: 10.1115/1.3140713. 2.1.1, 2.2.4
- Neville Hogan. Impedance Control: An Approach to Manipulation: Part I—Theory. *Journal of Dynamic Systems, Measurement and Control*, 107: 1–7, 1985b. 1.1, 1.2.1, 2.1.1
- Matthew Howard, David J Braun, and Sethu Vijayakumar. Transferring Human Impedance Behavior to Heterogeneous Variable Impedance Actuators. *IEEE Transactions on Robotics*, 29(4):847–862, 2013. 2.1.4
- Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006. ISSN 09252312. doi: 10.1016/j.neucom.2005.12.126. 2.1.6
- A.J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. *IEEE Intl. Conf. on Robotics and Automation*, pages 1398–1403, 2002. doi: 10.1109/ROBOT.2002.1014739. 2.1.6, 7.4.2

- Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural computation*, 25(2):328–73, 2013. ISSN 1530-888X. doi: 10.1162/NECO_a_00393. 2.1.6, 2.1.7, 7.4.2, 7.4.2, 7.4.2, 7.26
- M. Ito. On-line Imitative Interaction with a Humanoid Robot Using a Dynamic Neural Network Model of a Mirror System. *Adaptive Behavior*, 12(2):93–115, June 2004. ISSN 1059-7123. doi: 10.1177/105971230401200202. 2.1.6
- LA Jones. Kinesthetic sensing. in *Human and Machine Haptics*, 2000. 3.3.1
- Mrunal Kalakrishnan, Ludovic Righetti, Peter Pastor, and Stefan Schaal. Learning force control policies for compliant manipulation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4639–4644. IEEE, September 2011. ISBN 9781612844558. doi: 10.1109/IROS.2011.6095096. 2.1.8
- S. B. Kang and K. Ikeuchi. A robot system that observes and replicates grasping tasks. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1093–1099, 1995. ISBN 0-8186-7042-8. doi: 10.1109/ICCV.1995.466771. 2.1.3
- H K Khalil. *Nonlinear Systems, Third Edition*. 2002. ISBN 0130673897. doi: 10.1016/j.physa.2006.08.011. 2.2.1, 2.2.2
- M. Khansari-Zadeh and A. Billard. Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6):752–765, June 2014. ISSN 09218890. doi: 10.1016/j.robot.2014.03.001. 2.1.7, 6.5, 7.1
- Seyed Mohammad Khansari-Zadeh and Aude Billard. A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32(4):433–454, March 2012. ISSN 0929-5593. doi: 10.1007/s10514-012-9287-y. 6.5
- SM. Khansari-Zadeh and Aude Billard. Learning stable non-linear dynamical systems with Gaussian Mixture Models. *IEEE Transactions on Robotics*, 27: 1–15, 2011. 1.2.2, 2.1.6, 2.1.7, 3.3.3, 6.2.1, 6.4.1, 7.1, 7.6
- SM Khansari-Zadeh, 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 (RSS)*, 2014. 1.4, 4.4.2
- O Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986. 2.1.6
- O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, February 1987. ISSN 0882-4967. doi: 10.1109/JRA.1987.1087068. 2.1.1, 2.2.4, 2.2.4
- Seungsu Kim, Ashwini Shukla, and Aude Billard. Catching Objects in Flight. *IEEE Transactions on Robotics*, PP:1–17, 2014. ISSN 1552-3098. doi: 10.1109/TRO.2014.2316022. 2.1.6, 7.1
- Y Kishi, Zhiwei Luo, F Asano, and S Hosoe. Passive impedance control with time-varying impedance center. *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 1207–1212, 2003. 7.6

- Jens Kober and Jan Peters. Policy search for motor primitives in robotics. *Machine Learning*, 84(1-2):171–203, November 2010. ISSN 0885-6125. doi: 10.1007/s10994-010-5223-6. 2.1.6
- DE Koditschek. Robot planning and control via potential functions. *The International Journal of Robotics Review*, 1988. 2.1.6
- Petar Kormushev, Sylvain Calinon, and D.G. Caldwell. Robot motor skill coordination with EM-based reinforcement learning. In *IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3232–3237, 2010. 2.1.6
- Petar Kormushev, S. Calinon, and D.G. Caldwell. Imitation Learning of Positional and Force Skills Demonstrated via Kinesthetic Teaching and Haptic Input. *Advanced Robotics*, 25:581–603, 2011. 2.1.3, 2.1.4, 2.1.4
- Vasiliki Koropouli, Sandra Hirche, and Dongheui Lee. Learning and generalizing force control policies for sculpting. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1493–1498, October 2012. doi: 10.1109/IROS.2012.6385957. 2.1.3
- K. Kronander, S.M. Khansari-Zadeh, and A. Billard. Incremental Motion Learning with Locally Modulated Dynamical Systems. *Robotics and Autonomous Systems*, 2015. 1.4, 6.1
- Klas Kronander and Aude Billard. Online Learning of Varying Stiffness Through Physical Human-Robot Interaction. In *IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2012. 1.4, 3.1
- Klas Kronander and Aude Billard. Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction. *Transactions on Haptics*, 7(3):1–16, 2013. ISSN 1939-1412. doi: 10.1109/TOH.2013.54. 1.4, 2.1.8, 3.1, 3.3
- Klas Kronander, Mohammad Khansari-zadeh, and Aude Billard. Learning to control planar hitting motions in a minigolf-like task. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 710–717. IEEE, 2011. 1.4, 2.1.6, 2.1.8, 7.1
- D. Kulis, W. Takano, and Y. Nakamura. Incremental Learning, Clustering and Hierarchy Formation of Whole Body Motion Patterns using Adaptive Hidden Markov Chains. *The International Journal of Robotics Research*, 27(7):761–784, July 2008. ISSN 0278-3649. doi: 10.1177/0278364908091153. 2.1.7
- Tomas Kulvicius, Martin Biehl, Mohamad Javad Ain, Minija Tamosiunaite, and Florentin Wörgötter. Interaction learning for dynamic movement primitives used in cooperative robotic tasks. *Robotics and Autonomous Systems*, 61(12):1450–1459, 2013. ISSN 09218890. doi: 10.1016/j.robot.2013.07.009. 7.4.2
- E. Kunesch, F. Binkofski, and H.J. Freund. Invariant temporal characteristics of manipulative hand movements. *Experimental Brain Research*, 78(3):539–546, 1989. 3.3.1
- Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Learning by watching: extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822, 1994. ISSN 1042296X. doi: 10.1109/70.338535. 2.1.3

- Matteo Laffranchi, Nikos Tsagarakis, and D. G. Caldwell. A compact compliant actuator (CompAct™) with variable physical damping. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4644–4650, 2011. ISBN 9781612843865. doi: 10.1109/ICRA.2011.5979915. 2.1.1
- Steven M LaValle. *Planning Algorithms*. Cambridge University Press, 2006. ISBN 9780511546877. doi: 10.1017/CBO9780511546877. 2.1.6
- Alex X Lee, Henry Lu, Abhishek Gupta, Sergey Levine, and Pieter Abbeel. Learning Force-Based Manipulation of Deformable Objects from Multiple Demonstrations. *IEEE International Conference on Robotics and Automation*, 2015. 2.1.4, 8.2
- Christopher Lee and Yangsheng Xu. Online, interactive learning of gestures for human/robot interfaces. In *IEEE International Conference on Robotics and Automation*, number April, pages 2982–2987, 1996. 2.1.3
- Dongheui Lee and Christian Ott. Incremental kinesthetic teaching of motion primitives using the motion refinement tube. *Autonomous Robots*, 31:115–131, 2011. ISSN 0929-5593. doi: 10.1007/s10514-011-9234-3. 2.1.7
- Dongjun Lee and Ke Huang. Passive-set-position-modulation framework for interactive robotic systems. *IEEE Transactions on Robotics*, 26(2):354–369, 2010. ISSN 15523098. doi: 10.1109/TRO.2010.2041877. 7.3.3
- A Lemme, K Neumann, F.R Reinhart, and J.J Steil. Neurally imprinted stable vector fields. In *European Symposium on Artificial Neural Networks*, 2013. 2.1.6, 6.4.1
- A. Lemme, K. Neumann, R. F. Reinhart, and J. J. Steil. Neural learning of vector fields for encoding stable dynamical systems. *Neurocomputing*, 141: 3–14, 2014. ISSN 18728286. doi: 10.1016/j.neucom.2014.02.012. 2.1.6, 2.1.7, 6.5, 7.6
- A. Levas and M. Selfridge. A user-friendly high-level robot teaching system. *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, 1, 1984. ISSN 0163-5719. doi: 10.1109/ROBOT.1984.1087193. 2.1.3
- Miao Li, Hang Yin, Kenji Tahara, and Aude Billard. Learning object-level impedance control for robust grasping and dexterous manipulation. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6784–6791, May 2014. doi: 10.1109/ICRA.2014.6907861. 2.1.4
- Michael Y. Li and James S. Muldowney. On R.A. Smith’s autonomous convergence theorem. *Rocky Mountain Journal of Mathematics*, 25(1), 1995. 6.5
- Perry Y. Li and Roberto Horowitz. Control of smart exercise machines-part II: Self-optimizing control. *IEEE/ASME Transactions on Mechatronics*, 2(4): 248–258, 1997a. ISSN 10834435. doi: 10.1109/3516.653049. 2.1.8
- P.Y. Li. Adaptive passive velocity field control. *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, 2(June):774–779, 1999. ISSN 0743-1619. doi: 10.1109/ACC.1999.783145. 2.1.8
- P.Y. Li and R. Horowitz. Control of smart exercise machines. I. Problem formulation and nonadaptive control. *IEEE/ASME Transactions on Mechatronics*, 2(4), 1997b. ISSN 1083-4435. doi: 10.1109/3516.653048. 2.1.8

- P.Y. Li and R. Horowitz. Passive velocity field control of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 15(4):751–763, 1999. ISSN 1042296X. doi: 10.1109/70.782030. 2.1.8, 7.1, 7.6
- Chun-Shin Lin Chun-Shin Lin, Po-Rong Chang Po-Rong Chang, and J. Y. S. Luh J. Y. S. Luh. Formulation and optimization of cubic polynomial joint trajectories for mechanical manipulators. *1982 21st IEEE Conference on Decision and Control*, 21(0018), 1982. ISSN 0018-9286. doi: 10.1109/CDC.1982.268455. 1.2.1
- DT Lin, JE Dayhoff, and PA Ligomenides. Trajectory production with the adaptive time-delay neural network. *Neural Networks*, 8(3):447–461, 1995. 2.1.6
- H. Lipkin and J. Duffy. Hybrid Twist and Wrench Control for a Robotic Manipulator. *Journal of Mechanisms Transmissions and Automation in Design*, 110(2):138, 1988. ISSN 07380666. doi: 10.1115/1.3258918. 2.1.1
- J. Loncaric. Normal forms of stiffness and compliance matrices. *IEEE Journal on Robotics and Automation*, 3(6), 1987. ISSN 0882-4967. doi: 10.1109/JRA.1987.1087148. 2.1.1, 2.2.4
- T. Lozano-Perez. Robot programming. *Proceedings of the IEEE*, 71:821–841, 1983. ISSN 0018-9219. doi: 10.1109/PROC.1983.12681. 2.1.3
- Thomas Lozano-Perez, Matthew T. Mason, and Russel H. Taylor. Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research*, 1984. 2.1.5, 5.1
- Matthew T. Mason. Compliance and Force Control for Computer Controlled Manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6): 418–432, 1981. ISSN 0018-9472. doi: 10.1109/TSMC.1981.4308708. 2.1.1
- JR Medina, Dominik Sieber, and Sandra Hirche. Risk-sensitive Interaction Control in Uncertain Manipulation Tasks. In *IEEE International Conference on Robotics and Automation*, 2013. 2.1.4
- JR Medina, Tamara Lorenz, and Sandra Hirche. Synthesizing Anticipatory Haptic Assistance Considering Human Behavior Uncertainty. *IEEE Transactions on Robotics*, 31(1):180–190, 2015. 2.1.4
- S.A. Migliore, E.A. Brown, and S.P. DeWeerth. Biologically Inspired Joint Stiffness Control. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005. doi: 10.1109/ROBOT.2005.1570814. 2.1.1
- D. Mitrovic, S. Klanke, and S. Vijayakumar. Learning Impedance Control of Antagonistic Systems Based on Stochastic Optimization Principles. *The International Journal of Robotics Research*, 30(5):556–573, 2011. ISSN 0278-3649. doi: 10.1177/0278364910387653. 2.1.4
- RM Neal and GE Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer Netherlands, 1998. 2.1.7
- Klaus Neumann, Andre Lemme, and Jochen J. Steil. Neural learning of stable dynamical systems based on data-driven Lyapunov candidates. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1: 1216–1222, November 2013. doi: 10.1109/IROS.2013.6696505. 1.2.2, 2.1.7, 6.4.1

- Tetsuya Ogata, Shigeki Sugano, and Jun Tani. Open-end human–robot interaction from the dynamical systems perspective: mutual adaptation and incremental learning. *Advanced Robotics*, 19(6):651–670, January 2005. ISSN 0169-1864. doi: 10.1163/1568553054255655. 2.1.7
- Romeo Ortega, Antonio Loria, Per Johan Nicklasson, and Herbertt Sira-Ramirez. *Passivity-based control of Euler-Lagrange systems*. Springer-Verlag London, 1998. ISBN 9781849968522. 1.2.2
- R Osu and H Gomi. Multijoint muscle regulation mechanisms examined by measured human arm stiffness and EMG signals. *Journal of neurophysiology*, 81(4):1458–1468, 1999. ISSN 0022-3077. 2.1.4
- C Ott. *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, volume 49. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-69253-9. doi: 10.1007/978-3-540-69255-3. 2.2.4
- Christian Ott, Alin Albu-Schaffer, Andreas Kugi, Stefano Stramigioli, and Gerd Hirzinger. A passivity based cartesian impedance controller for flexible joint robots-part I: Torque feedback and gravity compensation. In *IEEE International Conference on Robotics & Automation*, pages 2659–2665, 2004. ISBN 0780382323. 2.1.1
- Christian Ott, Ranjan Mukherjee, and Yoshihiko Nakamura. Unified impedance and admittance control. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 554–561, 2010. ISBN 9781424450381. doi: 10.1109/ROBOT.2010.5509861. 2.1.1
- G Palli, C Melchiorri, and A De Luca. On the Feedback Linearization of Robots with Variable Joint Stiffness. In *International Conference on Robotics and Automation*, pages 1753–1759, 2008. ISBN 9781424416479. 2.1.1
- Hyeonjun Park, Ji-hun Bae, Jae-han Park, Moon-hong Baeg, and Jaeheung Park. Intuitive peg-in-hole assembly strategy with a compliant manipulator. In *International Symposium on Robotics (ISR)*, pages 1–5, October 2013. ISBN 978-1-4799-1173-8. doi: 10.1109/ISR.2013.6695699. 2.1.5, 5.7
- Peter Pastor, Ludovic Righetti, Mrinal Kalakrishnan, and Stefan Schaal. Online movement adaptation based on previous sensor experiences. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 365–371. IEEE, 2011. ISBN 9781612844558. 2.1.6, 7.4.2
- Barak Pearlmutter. Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1(2):263–269, 1989. 2.1.6
- G.A. Pratt and M.M. Williamson. Series elastic actuators. *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, 1, 1995. ISSN 0143991X. doi: 10.1109/IROS.1995.525827. 2.1.1
- J Quiñonero Candela, CE Rasmussen, J Quinonero-Candela, and Joaquin Qui. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005. 6.3.4
- M H Raibert and J J Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems Measurement and Control*, 102(2):126–133, 1982. 2.1.1

- Carl Rasmussen and Chris Williams. *Gaussian processes for machine learning*. MIT Press, 2006. 6.3.2
- E Rimon and D E Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992. ISSN 1042296X. doi: 10.1109/70.163777. 2.1.6
- DA Rosenbaum. *Human motor control*. Academic Press, 2009. 2.1.4
- L. Rozo, P. Jimenez, and C. Torras. Robot learning from demonstration in the force domain. In *International Joint Conference on Artificial Intelligence (IJCAI), Workshop on Agents Learning Interactively from Human Teachers*, 2011. 2.1.3
- Leonel Rozo, Sylvain Calinon, Darwin Caldwell, Pablo Jimenez, Carme Torras, and Pablo Jiménez. Learning Collaborative Impedance-based Robot Behaviors. In *AAAI Conference on Artificial Intelligence*, 2013. ISBN 9781577356158. 2.1.4, 8.2
- Wilson J Rugh and Je S Shamma. Research on gain scheduling. *Automatica*, 36:1401–1425, 2000. 2.1.2
- Jee Hwan Ryu, Dong Soo Kwon, and Blake Hannaford. Stable teleoperation with time-domain passivity control. *IEEE Transactions on Robotics and Automation*, 20(2):365–373, 2004a. ISSN 1042296X. doi: 10.1109/TRA.2004.824689. 2.1.2
- JH Ryu, DS Kwon, and Blake Hannaford. Stability guaranteed control: Time domain passivity approach. *IEEE Transactions on Control Systems Technology*, 12(6):860–868, 2004b. 2.1.2
- J.K. Salisbury. Active stiffness control of a manipulator in Cartesian coordinates. In *IEEE Conference on Decision and Control*, volume 19, pages 95–100, 1980. 2.1.1, 2.1.4
- EL Sauser, BD Argall, and Giorgio Metta. Iterative learning of grasp adaptation through human corrections. *Robotics and Autonomous Systems*, 60(1):55–71, January 2011. ISSN 09218890. doi: 10.1016/j.robot.2011.08.012. 6.1
- S Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, June 1999. ISSN 1879-307X. 2.1.3
- Stefan Schaal. Dynamic movement primitives—a framework for motor control in humans and humanoid robotics. In *The International Symposium on Adaptive Motion of Animals and Machines*, 2003. 1.2.2, 2.1.6
- Stefan Schaal, Auke Ijspeert, and Aude Billard. Computational approaches to motor learning by imitation. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 358(1431):537–47, March 2003. ISSN 0962-8436. doi: 10.1098/rstb.2002.1258. 1.1, 1.2.1, 2.1.4, 2.1.6
- G. Schoner. A dynamic theory of coordination of discrete movement. *Biological Cybernetics*, 63(4):257–270, 1990. ISSN 03401200. doi: 10.1007/BF00203449. 1.2.2
- G Schöner and J A Kelso. Dynamic pattern generation in behavioral and neural systems. *Science*, 239(4847):1513–20, 1988. ISSN 0036-8075. doi: 10.1126/science.3281253. 1.1, 1.2.2

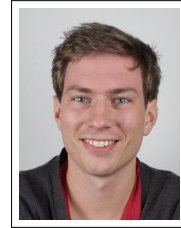
- Günter Schreiber, Andreas Stemmer, and Rainer Bischoff. The fast research interface for the kuka lightweight robot. *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010)*, pages 15–21, 2010. 7.5
- A. Segre and G. DeJong. Explanation-based manipulator learning: Acquisition of planning ability through observation. *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, 2, 1985. doi: 10.1109/ROBOT.1985.1087311. 2.1.3
- Luc P J Selen, David W Franklin, and Daniel M Wolpert. Impedance control reduces instability that arises from motor noise. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 29(40):12606–16, October 2009. ISSN 1529-2401. doi: 10.1523/JNEUROSCI.2826-09.2009. 1.1, 2.1.4
- Jeff S. Shamma and Michael Athans. Gain Scheduling: Potential Hazards and Possible Remedies. *Control Systems , IEEE*, 12(3):101–107, 1992. 2.1.2, 4.1
- A. P. Shon, K. Grochow, and R. P. N. Rao. Robotic Imitation for Human Motion Capture Using Gaussian Processes. In *5th IEEE-RAS International Conference on Humanoid Robots*, pages 129–134. Ieee, 2005. ISBN 0-7803-9320-1. doi: 10.1109/ICHR.2005.1573557. 2.1.3
- Ashwini Shukla and Aude Billard. Coupled dynamical system based arm-hand grasping model for learning fast adaptation strategies. *Robotics and Autonomous Systems*, 60(3):424–440, 2011. 2.1.6, 3.3.3
- P Sikka and BJ McCarragher. Stiffness-based understanding and modeling of contact tasks by human demonstration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 464–470, 1997. 2.1.4
- J.-J. E. Slotine. On the Adaptive Control of Robot Manipulators. *The International Journal of Robotics Research*, 6:49–59, 1987. ISSN 0278-3649. doi: 10.1177/027836498700600303. 2.1.1, 7.1
- Jean-Jaques Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice hall, 1991. 2.2.1, 2.2.2, 4.3.1, 4.3.1, 6.5
- RA Smith. Some applications of Hausdorff dimension inequalities for ordinary differential equations. *Proceedings of the Royal Society of Edinburgh: Section A*, (104A):235–259, 1986. 6.5
- Mingzhou Song and Hongbin Wang. Highly efficient incremental estimation of gaussian mixture models for online data stream clustering. *Defense and Security*, pages 174–183, March 2005. doi: 10.1117/12.601724. 2.1.7
- Stefano Stramigioli. *Modeling and IPC Control of Interactive Mechanical Systems - A coordinate-Free Approach*. Springer-Verlag New York, 2001. ISBN 3540372571. doi: 10.1007/11597018_1. 2.1.3, 2.1.8, 2.2.4
- Stefano Stramigioli, Cristian Secchi, Arjan J. van der Schaft, and Cesare Fantuzzi. Sampled data systems passivity and discrete port-Hamiltonian systems. *IEEE Transactions on Robotics*, 21:574–587, 2005. ISSN 15523098. doi: 10.1109/TRO.2004.842330. 2.1.2, 4.1, 7.3.3, 7.6
- Freek Stulp, Jonas Buchli, Alice Ellmer, Michael Mistry, Evangelos a. Theodorou, and Stefan Schaal. Model-free reinforcement learning of

- impedance control in stochastic environments. *IEEE Transactions on Autonomous Mental Development*, 4(4):330–341, 2012. ISSN 19430604. doi: 10.1109/TAMD.2012.2205924. 2.1.4
- C D Takahashi, Robert A Scheidt, and D J Reinkensmeyer. Impedance control and internal model formation when reaching in a randomly varying dynamical environment. *Journal of neurophysiology*, 86(2):1047–1051, 2001. ISSN 0022-3077. doi: citeulike-article-id:761586. 2.1.4
- M Takegaki and S Arimoto. A New Feedback Method for Dynamic Control of Manipulators. *Journal of Dynamic Systems Measurement and Control*, 103: 119–125, 1981. 2.1.2, 2.2.3, 2.2.4, 7.3.2
- E. Theodorou, J. Buchli, and Stefan Schaal. A generalized path integral control approach to reinforcement learning. *The Journal of Machine Learning Research*, 9999:3137–3181, 2010. 2.1.4
- Giovanni Tonietti, Riccardo Schiavi, and Antonio Bicchi. Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2005, pages 526–531, 2005. ISBN 078038914X. doi: 10.1109/ROBOT.2005.1570172. 2.1.1
- S.K. Tso and K.P. Liu. Hidden Markov model for intelligent extraction of robot trajectory command from demonstrated trajectories. *Proceedings of the IEEE International Conference on Industrial Technology (ICIT'96)*, 1996. doi: 10.1109/ICIT.1996.601593. 2.1.3
- C.P. Tung and A.C. Kak. Automatic learning of assembly tasks using a Data-Glove system. *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, 1, 1995. doi: 10.1109/IROS.1995.525767. 2.1.3
- Aleš Ude. Trajectory generation from noisy positions of object features for teaching robot paths. *Robotics and Autonomous Systems*, 11(2):113–127, 1993. ISSN 09218890. doi: 10.1016/0921-8890(93)90015-5. 2.1.3
- Aleš Ude, Christopher G. Atkeson, and Marcia Riley. Programming full-body movements for humanoid robots by observation. In *Robotics and Autonomous Systems*, volume 47, pages 93–108, 2004. doi: 10.1016/j.robot.2004.03.004. 2.1.3
- Arjan van der Schaft. *L2-gain and passivity techniques in nonlinear control*. Springer-Verlag London, 2000. ISBN 9781447111542. 1.2.2, 2.1.8, 2.2.2
- B. Vanderborght, a. Albu-Schaeffer, a. Bicchi, E. Burdet, D. G. 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. Van Damme, R. Van Ham, L. C. Visser, and S. Wolf. Variable impedance actuators: A review. *Robotics and Autonomous Systems*, 61(12):1601–1614, 2013. ISSN 09218890. doi: 10.1016/j.robot.2013.06.009. 2.1.1
- Sethu Vijayakumar, Aaron D’Souza, and Stefan Schaal. Incremental online learning in high dimensions. *Neural computation*, 17(12):2602–34, December 2005. ISSN 0899-7667. doi: 10.1162/089976605774320557. 6.3.4, 6.5

- Luigi Villani and Joris De Schutter. *Force Control, Chapter in handbook of robotics*. Springer, July 2008. doi: 10.1016/S0967-0661(01)00038-7. 2.1.1
- Tie Wang, Goran S. Dordevic, and Reza Shadmehr. Learning the dynamics of reaching movements results in the modification of arm impedance and long-latency perturbation responses. *Biological Cybernetics*, 85(6):437–448, 2001. ISSN 03401200. doi: 10.1007/s004220100277. 2.1.4
- D. Whitney. Historical perspective and state of the art in robot force control. *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, 2, 1985. ISSN 0278-3649. doi: 10.1109/ROBOT.1985.1087266. 2.1.1
- D. E. Whitney. Quasi-Static Assembly of Compliantly Supported Rigid Parts, 1982. ISSN 00220434. 2.1.5
- DE Whitney. Force feedback control of manipulator fine motions. *Journal of Dynamic Systems, Measurement and Control*, 99(2):91–97, 1977. 2.1.1
- D.E. Whitney and J.L. Nevins. What is the remote center compliance (RCC) and what can it do. *Proceedings of the 9th International Symposium on Industrial Robots*, pages 135–152, 1979. 2.1.1
- J Won and N Hogan. Stability properties of human reaching movements. *Experimental brain research*, 107(1):125–136, 1995. ISSN 0014-4819. 2.1.4
- Chenguang Yang, Gowrishankar Ganesh, Sami Haddadin, S. Parusel, A. Albu-Schaffer, and E. Burdet. Human-Like Adaptation of Force and Impedance in Stable and Unstable Interactions. *IEEE Transactions on Robotics*, 27(5): 918–930, 2011. 2.1.2, 2.1.4, 4.4.2
- Milos Zefran and Vijay Kumar. A Geometrical Approach to the Study of the Cartesian Stiffness Matrix. *Journal of Mechanical Design*, 124(1):30, 2002. ISSN 10500472. doi: 10.1115/1.1423638. 2.2.4

Klas Kronander

Chemin de l'Ochettaz 16
1025 St-Sulpice VD
Switzerland
☎ +41 79 102 21 26
✉ klas.kronander@gmail.com



Research Interests

- Robot Dynamics and Control
- Machine Learning
- Impedance Control
- Dynamical Systems
- Passivity-based Control Design
- Physical Human-Robot Interaction
- Learning from Demonstration
- Integrated learning and control

Education

- 2011–2015 **Doctoral School**, *EPFL*, Lausanne, Switzerland.
Enrolled in the Doctoral Program of Manufacturing Systems and Robotics. Courses on Robotics, Machine Learning and Signal Processing. Graduation in April/May 2015.
- 2006–2010 **MSc, Applied Physics and Electrical Engineering**, *Linköping University*, Sweden.
Specialization: Applied mathematics, Signal Processing and Automatic Control. The third year of studies was carried out at Institut National Polytechnique de Grenoble, France. Master Thesis title: Learning to Control Planar Hitting Motions in a Minigolf-like Task.

Professional Experience

- 2011–2015 **Doctoral Researcher**, *Learning Algorithms and Systems Laboratory, EPFL*, Lausanne.
Research on learning and controlling compliant manipulation tasks
Experience and Projects:
- Development of real-time control interfaces for two robot manipulators which are now used by a team of researchers.
 - Development of novel interfaces and learning algorithms for transferring skills to robots by demonstration.
 - Teaching activities in Robotic Practicals and Machine Learning MSc courses at EPFL. Supervision of 5 student projects.
- 2009 **Teaching Assistant**, *Linköping University*, Sweden.
Teaching assistant in calculus for first year engineering students.

Awards

- JTCF Novel Technology Paper Award My MSc thesis was submitted as a conference paper to the International Conference on Intelligent Robots and Systems, San Francisco, 2011 and was awarded the Japan Toy Culture Foundation Novel Technology Paper Award.
- NCCR Robotics best paper award My paper 'Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction' published in IEEE Transactions on Haptics was awarded the Swiss National Centre of Competence (NCCR Robotics) best PhD paper award 2014.

Publications

Journal Publications

- 2015 **Incremental Motion Learning with Locally Modulated Dynamical Systems**
K. Kronander, M. Khansari-Zadeh and A. Billard
Elsevier Robotics and Autonomous Systems, 2015, Vol. 70, pp. 52-62
- 2013 **Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction**
K. Kronander and A. Billard
IEEE Transactions on Haptics, 2013, Vol. 7, Iss. 3, pp. 367-380
- 2012 **Learning to Play Minigolf: A Dynamical System-based Approach**
M. Khansari-Zadeh, K. Kronander and A. Billard
Advanced Robotics, 2012, Vol. 26, Iss. 17, pp. 1967-1993

Peer-reviewed conference papers

- 2015 **Combined Kinesthetic and Simulated Interface for Teaching Robot Motion Models**
E. Cha, K. Kronander and A. Billard
Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2015, Kobe, Japan
- 2014 **Modeling robot discrete movements with state-varying stiffness and damping: A framework for integrated motion generation and impedance control**
M. Khansari-Zadeh, K. Kronander and A. Billard
Proceedings of Robotics, Science and Systems (RSS) X, 2014, Berkeley, CA, USA
- 2012 **Online Learning of Varying Stiffness through Physical Human-Robot Interaction**
K. Kronander and A. Billard
Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2012, St-Paul, MN, USA
- 2011 **Learning to Control Planar Hitting Motions in a Minigolf-like Task**
K. Kronander, M. Khansari-Zadeh and A. Billard
Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, San Francisco, CA, USA

Workshop proceedings and abstracts

- 2014 **Task Transfer via Collaborative Manipulation for Insertion Assembly**
K. Kronander, E. Burdet and A. Billard
Workshop on Collaborative Manipulation for Industrial Manufacturing, Robotics, Science and Systems (RSS) X, 2014, Berkeley, CA, USA
- 2014 **Incremental Motion Learning with Gaussian Process Modulated Dynamical Systems**
K. Kronander, M. Khansari-Zadeh and A. Billard
Workshop on Learning Plans with Context from Human Signals, Robotics, Science and Systems (RSS) X, 2014, Berkeley, CA, USA
- 2012 **Learning Joint Stiffness Variations from Demonstration**
K. Kronander and A. Billard
Workshop on Learning and Interaction in Haptic Robots, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, Villmoura, Portugal

Academic Activities

- 2015 Associate Editor for IROS 2015.
- 2015 Program Committee member for the ICRA 2015 workshop 'Compliant and Versatile Robot Control in Human Environments: Bridging the Gap between Learning and Control'.
- 2014 Main organizer of the workshop 'Compliant Manipulation: Challenges in Learning and Control' held in conjunction with IROS 2014 in Chicago.

2012-2015 Reviewer for numerous conferences and journals including IROS, ICRA, Humanoids, ICAR, IEEE Transactions on Robotics, IEEE Transactions on Mechatronics, Biological Cybernetics etc.

Skills

Technical Knowledge

Automatic Control Expert with several years of experience primarily with applications in robotics.

Machine Learning Expert knowledge and experience of a wide range of algorithms for regression and classification.

Programming Languages

C++ Expert. Real-time control and algorithm implementation.

Matlab Expert. Scientific computing, machine learning and control systems design.

Signal Processing Expert. Long experience with time-series modeling, filtering and forecasting.

Python Expert. Scientific computing, machine learning algorithms.

L^AT_EX Advanced. Experience with scientific typesetting.

Languages

Swedish **Mother tongue**

English **Fluent**

French **Advanced, level C1**

German **Intermediate, level B1**

Interests

Jazz, Paragliding, Skiing, Hiking