Multi-modal Skill Memories for Online Learning of Interactive Robot Movement Generation

by **Jeffrey Frederic Queißer**

Vorgelegt zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
Technische Fakultät, Universität Bielefeld
Juni 2018

Printed on permanent paper according to ∞ ISO 9706.
Gedruckt auf altersbeständigem Papier nach ∞ ISO 9706.

Acknowledgement

First, I would like to thank my primary supervisor Prof. Dr. Jochen Steil for his support and advice during the work on my PhD project, in particular, for his continuous support over the last five years.

Further, I would like to express my sincere gratitude to Prof. Dr. Barbara Hammer for providing me an excellent working environment as well as her strong scientific competence for the final phase of my studies.

I would like to thank Dr. Carola Haumann and Prof. Steil for support and organization of the CODEFOR project. The cooperation with Osaka University is a valuable asset, not only for my professional development, but also in terms of life experience.

Furthermore, I would like to express my thankfulness for the kind invitation of Prof. Minoru Asada to join his laboratory and to enjoy the excellent working environment. The work performed in his group under supervision of Dr. Hisashi Ishihara and Dr. Matthias Rolf became a vital part of my thesis.

Further special thanks are addressed to

- Dr. Yukie Nagai for fruitful discussions and for giving me the chance to present my work at her group meetings
- Dr. Felix Reinhart for his supervision during the first phase of my PhD
- Dr. Milad Malekzadeh and Dr. Alexander Schulz for the fruitful discussions and collaborations

...as well as all current/former members of CoR-Lab, the Asada Laboratory and the Machine Learning Group.

Finally, but not least, I want to thank all people I met at Bielefeld University. This thesis not only marks the end of my PhD studies of the last 4.5 years, it rather condenses almost 12 years of my life into $\sim$ 150 pages.

Abstract

Modern robotic applications pose complex requirements with respect to the adaptation of actions regarding the variance in a given task. Reinforcement learning can optimize for changing conditions, but relearning from scratch is hardly feasible due to the high number of required rollouts. This work proposes a parameterized skill that generalizes to new actions for changing task parameters. The actions are encoded by a meta-learner that provides parameters for task-specific dynamic motion primitives. Experimental evaluation shows that the utilization of parameterized skills for initialization of the optimization process leads to a more effective incremental task learning. A proposed hybrid optimization method combines a fast coarse optimization on a manifold of policy parameters with a fine-grained parameter search in the unrestricted space of actions. It is shown that the developed algorithm reduces the number of required rollouts for adaptation to new task conditions. Further, this work presents a transfer learning approach for adaptation of learned skills to new situations. Application in illustrative toy scenarios, for a 10-DOF planar arm, a humanoid robot point reaching task and parameterized drumming on a pneumatic robot validate the approach.

But parameterized skills that are applied on complex robotic systems pose further challenges: the dynamics of the robot and the interaction with the environment introduce model inaccuracies. In particular, high-level skill acquisition on highly compliant robotic systems such as pneumatically driven or soft actuators is hardly feasible. Since learning of the complete dynamics model is not feasible due to the high complexity, this thesis examines two alternative approaches: First, an improvement of the low-level control based on an equilibrium model of the robot. Utilization of an equilibrium model reduces the learning complexity and this thesis evaluates its applicability for control of pneumatic and industrial light-weight robots. Second, an extension of parameterized skills to generalize for forward signals of action primitives that result in an enhanced control quality of complex robotic systems. This thesis argues for a shift in the complexity of learning the full dynamics of the robot to a lower dimensional task-related learning problem. Due to the generalization in relation to the task variability, online learning for complex robots as well as complex scenarios becomes feasible. An experimental evaluation investigates the generalization capabilities of the proposed online learning system for robot motion generation. Evaluation is performed through simulation of a compliant 2-DOF arm and scalability to a complex robotic system is demonstrated for a pneumatically driven humanoid robot with 8-DOF.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Despite the tremendous technological development in the field of robotics and movement generation, dealing with unstructured environments, e.g. as faced for household applications, is still extremely challenging for robotic systems. Most application areas of autonomous robotics are still limited to classical repetitive industrial tasks like painting, welding, pick-and-place, and packaging. Typically, these tasks are characterized by predefined environments and a limited variance of the task. However, advances in material science and new actuator concepts improved the mobility and resulted in systems with the potential to be applied to more general tasks. One of the limiting factors of a more versatile application of robots is the lack of control methods that allow to cope with complex environments and complex robot systems. Robot task execution is often specialized for a certain task and lacks flexibility to generalize to changing task configurations. As an example, consider the task of opening a door. Despite being presumably easy, current research, e.g. [Jain et al., 2010; Endres et al., 2013; Nemec et al., 2017], and robotic challenges [Guizzo and Ackerman, 2015] show that this is still a challenging task for robotic systems. Mastering the skill of opening a door incorporates various factors that modulate the action of the robot for successful task completion. In this thesis, it is assumed that the task parameterization defines all factors that are relevant for successful task execution. For this example, the task parameterization can include the relative position between the robot and the door handle, the relative position of the handle and the joints of the door, and the shape of the handle. Further, the task parameterization can encode variable interaction forces, like the amount of force that is necessary to press the door handle or the friction of the joints. Although the robot may have seen previous situations during a training phase for a set of observed task parameters, actions have to be generalized for each unseen task instance. Such real world-tasks are performed in a complex environment that requires costly online executions of actions for optimization. Thus, the generalization from a low number of successful

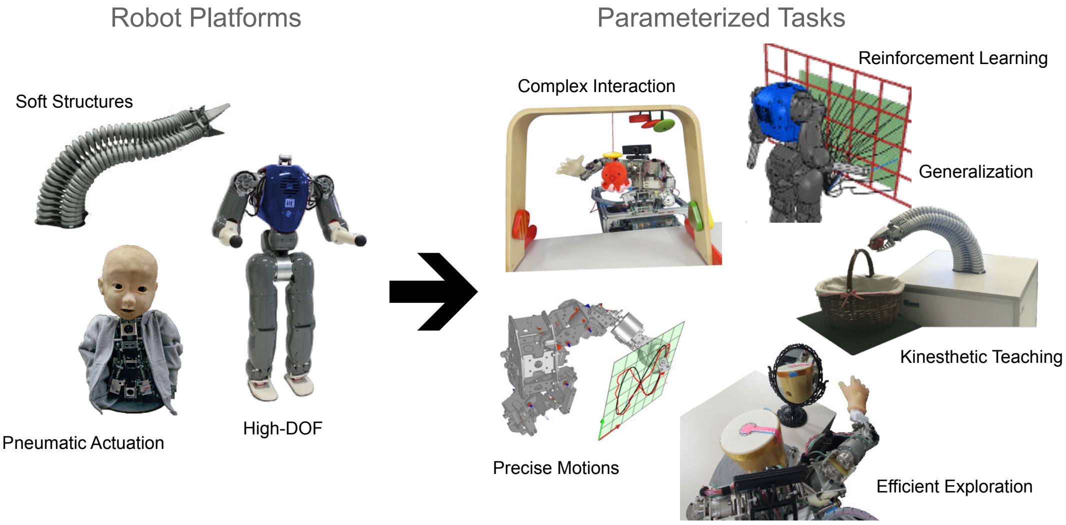


Figure 1.1: Presentation of the scope of this thesis.

actions becomes important.

Dealing with the variability of tasks is just one of many challenges. Leaving the structured environment makes high demands on the robot's structural properties as well. A new generation of light-weight actuators improves the weight-to-payload ratio, which makes more flexible applications possible and enhances the mobility of robot systems. Besides the lighter weight of the robot, introducing compliant elements and improved sensor capabilities leads to an enhanced safety for human-robot-interaction and allows sharing the workspace between humans and robots. Due to the enhanced safety, robot programming by interactive teaching, collaborative work and learning by exploration becomes feasible. But compliance and light-weight robot structures reduce the stiffness of the actuator and introduce model uncertainties that are difficult to handle. The Kuka-DLR light-weight arm [Hirzinger et al., 2002], for example, requires an additional vibration compensation in the joint controller due to structural deformations caused by a reduced stiffness [Albu-Schäffer et al., 2007].

In consideration of the growing complexity of the control of robotic systems that operate in application areas that introduce variability as well as interaction with humans and the environment, major bottlenecks for sophisticated action generation are generalization capabilities and robustness to perturbations. Classical control concepts struggle with the high complexity of the robot and the environment as well as a high variability regarding tasks because they rely on precise model-based control.

Those problems motivated research on biologically inspired concepts of motor control and actuator design. Skill learning of humans follows a fundamentally different concept compared to classical methods of robot task execution. As an example, the way humans learn to walk shows that skill learning does not rely on a simplification of the task. As shown in the *"developmental motor milestones"* of Adolph and Robinson [2013], prior to be able to stand and walk alone, children undergo a cruising phase in which they perform locomotion supported by grasping objects for support. This behavior emphasizes the differences to classical robot control in which, first, locomotion without the dynamics of additional contact points is learned and, later, further complexity is introduced in a successive extension of the dynamics model.

This poses the question of how the brain can handle the high complexity of successful skill acquisition?

One important biologically motivated concept for a reduction of the complexity of motion generation are motion primitives [Mussa-Ivaldi and Bizzi, 2000]. Motion primitives help to break down the complexity of motor control to action or goal directed motions that are considered as basic building blocks of longer actions. Recent research, as an example, demonstrates that event sequences based on only 8 atomic action primitives are sufficient for a compact description and identification of complex tasks such as *preparing breakfast* or *cutting and stirring milk* [Aksoy et al., 2015, 2016].

Biologically motivated architectures that aim for morphological computation are a further example of simplifying the control of a complex body. The term morphological computation can be described as "Offloading the computation from the brain to the body", as stated by Müller and Hoffmann [2017]. Classical robotic systems are built to support their representation by a dynamics model and this lays the foundation for high-level skill learning in contrast to biological systems that incorporate complex morphologies with over 600 skeletal muscles [Yin et al., 2012]. Nevertheless, research on biologically inspired robotic architectures reveals interesting concepts like the minimization of energy usage [Haq et al., 2011; Roozing et al., 2016], passive walking without control [McGeer, 1990] as well as a high chance for arm movements that result in opening a door by random exploration in the motor space [Hosoda et al., 2012]. The aforementioned arguments promote the view that musculoskeletal systems are optimized by evolutionary pressure for tasks solving, rather than to be precisely modeled by the human brain, including their complete dynamical properties. Consequently, the application of classical control schemes on those biologically inspired robots results in a poor performance, since modeling of the actuator dynamics and its interaction in a high-dimensional state space is not feasible. High-level task learning relies on the exact execution of motions and is therefore prone to inaccuracies on these architectures.

The discussed challenges of skill learning with respect to the variability of the environment, complex morphological structures and dynamics of the robot including interaction with the environment, yield motivation for the work presented in this thesis. Under the assumption that complex motor skills are composed of basic movement primitives, efficient learning of parameterized motion primitives that can be executed on complex robot systems is investigated. The challenges addressed in this thesis can be classified into two main scopes: First, online learning of a representation of the parameterized movements; Second, the execution of movements on real robotic systems that face complex dynamics. The complex dynamics properties can be caused by e.g. the robot's structure as well as interactions with humans for teaching or interactions with objects during manipulation. In the following, both scopes, which are addressed in this work, will be introduced in detail.

Movement Generalization

Approaches of motion representation through dynamic motor primitives lack the ability to flexibly integrate different levels of representation and modalities. Whereas impressive progress has been made to optimize such movement generation [e.g. Stulp and Sigaud, 2013; Kober and Peters, 2010], policy search has to be applied in a high dimensional space of parameters of the motor primitives. Searching in this high-dimensional parameter space requires a large number of samples and is therefore not applicable for online robotic systems, since the generation of online training samples is usually very costly. Explicit parameterization of higher-level goals in the search space as in [Ude et al., 2010; Kober and Peters, 2010], e.g. to go through via-points, is possible but inflexible and cannot easily be relearned. Again, optimization requires a large number of samples and typically relies on reward-weighted averaging of so called rollouts. These are executions of the movement policy under random perturbations that require a very careful parameterization of the reward and costly executions of the trial movements on the real robot. This scheme cannot easily be extended to respect multi-modal or higher-order goals.

Recent work introduces parameterized skill representations inspired by general motor schemas [Schmidt, 1975], which propose a motor program that is modulated by a memory structure. The memory links high-dimensional motor primitives to a low-dimensional embedding that represents high-level task descriptions. Ijspeert et al. [2013] propose models for action generation based on dynamic motion primitives and perceptual coupling. Further work extends this idea and introduces parameterized skills to perform a generalization of action primitives based on a high-level task description [Kober and Peters, 2010; Silva et al., 2012; Kober et al., 2012; Reinhart and Steil, 2014; Baranes and Oudeyer, 2013; Mülling et al., 2013; Silva et al., 2014]. To tackle the problem of multi-modal representations in movement control, Reinhart and Steil [2015] have introduced a parameterized skill memory through an associative dynamical systems approach. It is based on earlier work on associative multi-modal memories [Emmerich et al., 2013] and results in a significant reduction of reward episodes.

Complex Dynamical Properties

Previous approaches for parameterized task representations focus on the representation of the kinematic properties of the task, e.g. trajectories in joint or end effector space. It is assumed that a low-level controller exists that executes the estimated motions of the robot. But the application of parameterized skills on real robotic platforms has to face model uncertainties caused by a complex structure of the robot, compliance, and dynamic effects like friction, noise or delays. Classical approaches assume the existence of a high-quality dynamics model of the robot. Such a model allows an estimation of motor signals that are supposed to result in the desired motions in combination with a feedback controller that compensates for model errors and disturbances. Usually, a parameterized dynamics model is used that covers unknown properties, such as friction or weight of the rigid parts of the robot. But with an increasing complexity of robot systems, estimation of the model becomes more difficult. In particular, highly compliant actuators with continuously deformable parts, such as light-weight, pneumatic or soft robots, are difficult to model. To enhance the quality of control, hybrid models have been proposed. They combine analytical modeling techniques with data-driven approaches, e.g. machine learning. As en example, a function approximator can be trained to estimate model errors as proposed by Reinhart et al. [2017a]. In [Shareef et al., 2016], it is assumed that learning the model errors is easier (less jerk/curvature, stronger regularization) than learning the complete dynamics of the robot from scratch. But still, application of learning approaches remains difficult due to the large state space.

Additionally, the interaction in a complex and changing environment has to be considered for low-level control of the actuators as well. Interactions with the environment, like multiple contact points, result in a significant increase of the model complexity, as the dynamics of the environment has to be considered as well. This becomes even more difficult in case manipulation of objects takes place that involves complex dynamic properties caused by fluids, plasticity or even further completely unmodeled dynamics.

1.2 Problem Statement

Motivated by the preceding discussion, the problem statement for this thesis will be formulated. The central aspect of this thesis is the extension of previous work on parameterized skills. The chosen task representations play a crucial role to infer flexible generalizations of learned movements that can be adapted to new task situations. To be applicable to real robot scenarios, a framework is required that allows for online learning, i.e. application on online systems, as well as incremental learning in-the-loop. This is necessary, because the variability of a given application area cannot be covered in a simulation and must be explored online. Therefore, an adaptation to the current task is required by gathering a primarily small number of training samples. The success of the approach can be directly measured in terms of required trials a robot has to execute for skill acquisition. Implementation shall demonstrate the applicability for systems with many degrees-of-freedom as well as real time and online constraints. Applications aim for complex robot systems that pose further challenges for a successful skill execution, e.g. no model-based control available, sensory noise, no rigid body structure, compliance or long delays (or poor quality). Experimental evaluation includes the generalization capabilities for real world scenarios and interaction with the environment. This requires the adaptation of movements to new task instances based on training instances shown by a human tutor or gathered by reinforcement learning.

1.2.1 Research Questions & Contribution

In the following, the key aspects and the related research questions that will be addressed throughout this work will be discussed. The presentation of the key aspects is ordered by their occurrence in the title of this thesis:

Multi-Modality The generalization of actions for parameterized tasks is based on high-level information that describes the variability of the task. This requires the integration of different modalities including parameterizations that influence the shape of the required trajectory like obstacle positions or the target position and rotation. Further modalities that do not influence the shape but the dynamic properties of the task could be defined. Those properties could include weight of payload, execution speed, and physical properties during interaction, e.g. friction.

A further challenge arises from the question of how previous knowledge can be reused (for adaptation to changes of the task configuration). Skill learning is a time-consuming process that requires human demonstrations or optimization by trial-and-error. Adaptation instead of relearning of an action repertoire for new task conditions could be beneficial to speed up skill exploration. The following research questions address the aforementioned challenges:

- **RQ1**: How to achieve a multi-modal representation for action generalization?
- **RQ2**: How to adapt previously learned skills to an altered perception or across modalities?

Research question **RQ1** is addressed in Chapter 2 by introducing a conceptual framework for parameterized skill learning that is used throughout this work. Implementation and experimental evaluation for kinematic task representations is presented in Chapter 3. Transfer learning of skills is investigated in Chapter 4 and demonstrates the transfer of the skill of drumming between different modalities (**RQ2**). This thesis presents a novel transfer learning method for nonlinear regression tasks based on previous work for transfer of classification tasks [Prahm et al., 2016; Paaßen et al., 2018]. It is demonstrated that the transfer of the skill is significantly more efficient than relearning from scratch.

Skill Memories The parameterized skill is a memory structure that is used to generalize from observed task parameterizations to actions. Each training sample is acquired by multiple executions of perturbated actions. Gathering training data is costly, since complex scenarios impede simulation-based optimization and a high number of executions is necessary. One option to reduce the number of executed trials is the improvement of the generalization capabilities of the memory which results in a lowered number of required training samples. A further option is the reduction of required trials per task instance by an improved optimization process.

The following related research questions are addressed in this thesis:

- **RQ3**: How to improve generalization capabilities of the skill representation?
- **RQ4**: How to achieve an efficient estimation of solutions for unsolved task instances?

In Chapter 3, an improved representation of parameterized skills by an additional optimization constraint is proposed which addresses **RQ3**. This thesis refers to the constraint for optimization as *regularization of reward*. The regularization of reward penalizes the distance of solutions of the optimization process to the current estimation of the parameterized skill. Further, an incremental bootstrapping of the parameterized skill is proposed in Chapter 3 for a reduction of the required rollouts for skill learning. Experimental evaluation shows a significant reduction of the number of required rollouts in simulation of a 2D planar arm and a point reaching scenario of the upper body of a humanoid robot. Further evaluations demonstrate the bootstrapping for a drumming task on a pneumatically driven robot platform. To utilize previous knowledge for a reduction of the search space for optimization (**RQ4**), an algorithmic extension of the CMA-ES algorithm to multiple spaces is proposed in Chapter 4. The benefits of the proposed optimization in the space of policies and the space of task parameterizations is evaluated on toy data and on various robotic platforms.

Online Capabilities The proposed framework in Chapter 2 makes an effective integration of existing supervised online learning methods possible. A further challenge that is addressed in this thesis is the integration of the presented methods into an online system, including: 1) state-of-the-art optimization algorithms for efficient policy optimization; 2) the previously discussed regularization of the reward; 2) the bootstrapping process of the memory; 3)the optimization in hybrid spaces. These challenges motivate the research question:

- **RQ5**: How to implement a complex skill learning framework on an online system?

RQ5 is addressed by the following scenarios that demonstrate the applicability of the proposed methods of this thesis for real robotic applications:

1. Learning to drum for variable target positions on a pneumatically driven humanoid robot platform, Figure 1.1-②.
2. Kinesthetic teaching on a soft continuum trunk-shaped robot, Figure 1.1-①.
3. Complex interaction with a baby toy on a pneumatically driven humanoid robot platform, Figure 1.1-②.

Interaction Besides the question of how robots can interact with humans for learning or cooperative work, complex scenarios require the interaction with the environment as they are supposed to manipulate objects. Therefore, this thesis poses the following research questions:

- **RQ6**: How to achieve kinesthetic teaching on highly compliant robotic systems with unknown dynamic properties for complex task learning?
- **RQ7**: How to employ action generation that interacts with or manipulates the environment?

Research question **RQ6** is addressed in Chapter 5. Instead of an approximation of the complete dynamics of the robot, this thesis proposes to utilize a much simpler equilibrium model of the robot. The equilibrium model represents the relation of motor signals for static postures of the actuator with zero velocity and zero acceleration. Demonstration of the feasibility is performed on a pneumatically actuated trunk-shaped robot, a pneumatically actuated humanoid robot, and in a simulation of an industrial light-weight robot. To capture the interaction with the environment (**RQ7**) a complex scenario was developed, as presented in Chapter 6. In this scenario, a humanoid robot is intended to pull a toy that is attached via a rope to a spring mechanism. The pneumatic actuation of the robot and the interaction with the toy impede a precise control of the robot. Neither a model of the robot, nor a model of the interaction is available and successful task execution is not possible. This thesis shows that the integration of the dynamics representation into the parameterized skill (Chapter 2) allows the system to successfully master the given task.

Robot Movement Generation Task parameterized action generation focuses on the generalization of required joint angle trajectories or descriptions of end effector movements for successful task execution. Task execution on complex robots becomes difficult, since an optimal low-level controller for execution of the estimated trajectories is assumed to be available. This thesis addresses model-free control of complex robotic systems by the following research question:

- **RQ8**: How to execute motions on robots that have complex dynamics properties without the availability of model-based control?

In comparison to model-based control, this thesis proposes a task based generalization of forward signals. In the same way as for the kinematic representation of tasks by a parameterized skill, forward signals are generalized and support the feedback controller and to minimize the tracking error of the joints. This allows a representation of the dynamic properties of the robot in relation to the complexity of the given task instead of the complexity of the robot system and provides a solution to **RQ8**, as presented in Chapter 6.

1.3 Related References of the Author

Large parts of the thesis have been presented to an international audience in the following journal, conference, and workshop publications, all of which have been peer reviewed if not otherwise noted:

Journal Articles:

- Rolf, M., K. Neumann, J. F. Queißer, F. Reinhart, A. Nordmann, and J. J. Steil
  2015. A multi-level control architecture for the bionic handling assistant. *Advanced Robotics*, 29(13: SI):847–859

- Queißer, J. F. and J. J. Steil
  2018. Bootstrapping of parameterized skills through hybrid optimization in task and policy spaces. *Frontiers in Robotics and AI*, 5(49)

- Malekzadeh, M. S., J. F. Queißer, and J. J. Steil
  [Submitted]. Control of bionic handling assistant robot by learning from demonstration. *Advanced Robotics*

Conference Contributions:

- Queißer, J. F., K. Neumann, M. Rolf, R. F. Reinhart, and J. J. Steil
  2014. An active compliant control mode for interaction with a pneumatic soft robot. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pp. 573–579

- Queißer, J. F., R. F. Reinhart, and J. J. Steil
  2016. Incremental bootstrapping of parameterized motor skills. In *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Pp. 223–229

- Balayn, A., J. F. Queißer, M. Wojtynek, and S. Wrede
  2016. Adaptive handling assistance for industrial lightweight robots in simulation. In *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, Pp. 1–8

- Malekzadeh, M. S., J. F. Queißer, and J. J. Steil
  2017b. Imitation learning for a continuum trunk robot. In *Proceedings of the 25. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. ESANN 2017*, M. Verleysen, ed., Pp. 335–340. Ciaco

- Queißer, J. F., H. Ishihara, B. Hammer, J. J. Steil, and M. Asada
  2018. Skill memories for parameterized dynamic action primitives on the pneumatically driven humanoid robot child affetto. In *Joint IEEE International*

Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), Tokyo, Japan. IEEE

- Schulz, A., J. F. Queißer, H. Ishihara, and M. Asada
  2018. Transfer learning of complex motor skills on the humanoid robot affetto. In *Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE

Workshop and Symposium Contributions:

- Malekzadeh, M. S., J. F. Queißer, and J. J. Steil
  2015. Learning from demonstration for bionic handling assistant robot. In *IROS 2015 Workshop - New Frontiers and Applications for Soft Robotics*, Pp. 101–107
- Queißer, J. F. and J. J. Steil
  2016. Incremental bootstrapping of parametrized skill memories. In *DGR Days 2016*, D. D. 2016, ed., P. 14
- Malekzadeh, M. S., J. F. Queißer, and J. J. Steil.
  2016. Learning the end-effector pose from demonstration for the bionic handling assistant robot. In *9th International Workshop on Human-Friedly Robotics*

1.4 Funding Acknowledgements

The following institutions and associated grants are gratefully acknowledged:

- This research was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG) - https://www.cit-ec.de
- J. Queißer received funding from the CODEFROR project (FP7-PIRSES-2013-612555) - https://www.codefror.eu/

1.5 Organization of the Dissertation

The remainder of this work is structured as follows.

Chapter 2 introduces related work on skill learning. This includes a chronologically structured discussion on the fundamental theories related to motor skill learning in Section 2.1. The presented work lays the foundation for current concepts of robot skill learning and this thesis. Subsequently, an overview of robotic systems and their control approaches is given. Unsolved challenges of robot control motivate the discussion on biologically inspired concepts of motor control. Finally, an overview of recent primitive-based methods for parameterized skills is given. The second part, Section 2.2, refers to the previously discussed models of motor learning and their limitations to propose a novel framework for skill learning. It is followed by a discussion on the implementations of specific modules of the proposed framework and an overview of the robotic platforms and used data sets for evaluation, as presented in Section 2.2.2. The following chapters will refer to this framework and present work that implements, evaluates or extends specific parts of the framework.

Chapter 3 proposes a regularization of the reward function as well as a bootstrapping mechanism for efficient skill learning based on the parameterized skills. The bootstrapping as presented in Section 3.2 aims at a reduction of the required rollouts that are necessary for optimization of unsolved task instances. An additional regularization of the reward (Section 3.3) enhances the generalization capabilities and reduces thereby the number of required optimized task instances. Experimental evaluation on toy data, simulated robotic actuators and real robot systems demonstrate the feasibility of the proposed approach.

Chapter 4 proposes methods for a more efficient skill learning based on a reduction of the search space for policy optimization. First, a hybrid optimization is proposed in Section 4.2 that combines optimization in policy and task space. Second, transfer learning for the adaptation of skills to a changing perception is investigated in Section 4.3. Experimental evaluation is performed on toy data, simulations and on real robotic experiments.

Chapter 5 presents methods for enhanced control of highly compliant robotic systems (Section 5.2) without availability of analytical models. The combination of learned equilibrium models, estimated by supervised learning, and classic control is proposed to enhance joint level control. Additionally, a compliant control mode is introduced that facilitates kinesthetic teaching. Evaluation is performed on a pneumatically actuated trunk-shaped robot (Section 5.3.1), the upper torso of a pneumatically actuated robot child (Section 5.3.2) and an industrial light-weight robot (Section 5.3.3).

Chapter 6 argues for generalization of forward signals in relation to high-level task parameterizations to overcome limitations of the equilibrium-based control of Chapter 5. The generalization of forward signals is supposed to support the low-level controller and allows learning of complex skills on complex robotic systems. First, an evaluation of a parameterized trajectory tracking task is performed in Section 6.2. Further, the method is applied to a complex interaction scenario as demonstrated in Section 6.3.2. This final scenario includes kinesthetic teaching, incremental optimization, generalization of joint-angle trajectories as well as forward signals for new task instances and a complex interaction with the environment. The experimental platform is the upper torso of a pneumatically driven humanoid robot child.

Chapter 2

Skill Represenation & Skill Learning

Chapter Overview *The first part of this chapter will introduce work related to parameterized skill learning: First, an overview of the historical background of theories of skill learning will be presented. Basic concepts that are related to motor skill learning will be introduced. Second, an overview of control approaches for robotic manipulators will be given. Problems caused by modern robotic configurations, which include compliant elements attached to the robot, impedance modes during operation and dynamic environments, will be discussed. Third, biologically motivated concepts for motor control will be presented that support complex action generation. Humans and animals developed a complex musculoskeletal morphology and are at the same time able to perform complex actions. Fourth, an overview of current frameworks for robot skill learning will be presented.*

The second part of this chapter will introduce a novel skill learning framework. This includes an overview, a formal definition and a comparison to related work. Successively, the second part will discuss details of the proposed skill learning architecture: the memory component responsible for generalization, signal encoding of motor commands that are sent to the low-level controller, the refinement of actions as well as robotic platforms and data sets that have been used for evaluation.

2.1 Background: From Theories of Motor Control to High-Level Skill Learning

The underlying concepts of current methods for robot skill learning go back to early theories of motor skill learning. Therefore, this section will give a chronological overview of concepts related to this thesis. This includes multi-modal representations, variables that influence motor learning, development of motor control theory, evaluation of skills, reinforcement learning, complexity of high-DOF, open- or closed-loop control and parameterized motor programs. The discussion is followed by a presentation of robotic systems that include classical, compliant and soft robots. As argued in the following, highly compliant and soft robots are difficult to control, which motivates the successive overview over biologically concepts of motor control. The final section presents recent architectures for high-level skill learning.

Origins of Parameterized Skill Learning:
Theories of Motor Skill Learning Early works on the theory of motor control have been conducted by the physiologist Sir. Charles Sherrington (1857-1952). He investigated basic mechanisms for neural control and movement generation and introduced the concept of common pathways for muscle activations [Sherrington, 1906; Burke, 2007]. Nerve impulses from different sources, like the brain, reflexes, and sensory receptors, form single spinal columns and become a unified signal for muscle groups. His work can be interpreted as an early description of the concept of multi-modal signal integration for action execution.

The early phase of the development of theories of motor skill learning was driven by the field of psychology. Wilhelm Wundt (1832-1920), the founder of the first experimental laboratory, established psychology as a legitimate science, separate from philosophy. He proposed studies for reaction time experiments [Wundt et al., 1907] to investigate variables that influence motor learning that are still common today. The experiments investigated variables like perception, sensation, and attention as discussed by Edwards [2010].

One of the first descriptions of motor control was given by the response-chaining hypothesis by William James. He introduced the idea of open-loop control for motor learning as an adaptation of reflexes [James, 1890].

William L. Bryan and Noble Harter performed studies about learning motor skills of patterns for generation of telegraph messages. Evaluation was performed by estimation of learning rates of individuals [Bryan and Harter, 1897]. Further analysis of motor control and learning was performed by Robert S. Woodworth by investigating the accuracy of voluntary movements [Woodworth, 1899]. The Law-of-Effect, attributed to Edward Thorndike (1874-1949), describes learning based on stimuli and their responses in a similar vein as the concept of reinforcement learning [Thorndike, 1898; Lattal, 1998]. Rewarded behaviors are more likely to be repeated, compared to punished ones, which are preferred to be avoided. Additional, he was involved in the introduction of the notion *transfer of practice*, later known as *Transfer of Learning*, dealing with the question of how to transfer knowledge gained by learning from one context to a similar one [Woodworth and Thorndike, 1901].

Nikolai Bernstein's research investigated how the brain controls the movements of the body and his research resulted in the formulation of the *Degrees of Freedom* (DOF) problem, which refers to the number of ways that components of a system are free to vary [Bernstein, 1967]. He argued that the redundancy of patterns on the cellular level of motor control can reach up to millions and the brain is not able to control them for complex skills. To address the problems of high-DOF, a reduction of the control complexity by a freezing of single DOFs and muscle *synergies* have been discussed. Muscle *synergies* are given by co-activation of muscles commanded by a single neural signal and they represent a simple mechanism for dimension reduction. As noted by Edwards [2010], the work of Bernstein was first published much earlier (1920-1930) in Russia, before it was translated in 1967.

The following World War II (1939-1945) influenced the research of psychologists driven by the need to train military personnel. Due to research that supports military institutions, e.g. selecting personnel for the air force, many tests for evaluation of motor and perceptual abilities have been developed. During that time, Clark L. Hull worked on a general theory of learning that promoted learning as a result of several factors that determine the likelihood of a specific behavior to occur [Hull, 1952]. Those factors include a drive reduction as an implicit encoding of a goal, a motivation by a reward, inhibition due to the absence of reward and prior experience. But his theory was too general and not adequate to describe processes and variables involved in motor learning in detail, as discussed in [Krahe, 1999].

In the following years, cognitive learning theories gained more attention. The information-processing approach, motivated by computational metaphor, lead to research aims different from task-based approaches, like the neural control of simple movements. This motivated the concept of a closed-loop theory of motor learning [Adams, 1971]. This work was motivated by closed-loop control of the servotheory of engineering, as outlined in [Adams, 1987].

An open loop controller promotes a central system that contains all or partial information necessary for movement generation. Such a centralized control scheme was the motivation for the concept of an activated program that is responsible for the generation of movements and a reduction of the importance of feedback information. But generalization would be poor if a system would have to learn a new motor program for every movement needed and additionally, endless storage would be required. To overcome the problems of motor programs, Richard A. Schmidt proposed the general motor schema [Schmidt, 1975]. His work introduces the notion of a generalized motor program (GMP), an abstract memory structure that generates responses for a movement class based on a parameterization. As an example, a single motor program would be responsible for various styles of movements that result in jumping: fast or slow, high or long, one-legged or two-legged. Later works extended this concept to focus on goal-oriented actions instead of movements as discussed in [Mulder and Hulstyn, 1984; Krahe, 1999].

The aforementioned concepts lay the foundation for the work presented in this thesis. In the following, task execution and skill learning on robotic platforms will be discussed.

**From Classical Robots to Soft Actuators:
Robotic Systems and their Control Approaches** One of the most prominent control modes for current robotic applications is position control. Position control on joint level was already a component of the first robotic system that was used for

automation in factories, the UNIMATE robot, described in the patent "Programmed Article Transfer" [Devol Jr., 1954]. The robot had to repeat the execution of a given target trajectory to fulfill a desired task, discrete joint positions, that define postures of the robot have been read as a temporal sequence from a magnetic memory. Since then, an astounding development in the field of robot control lead to a multitude of advanced control concepts. Modern robotic systems are able to operate in different coordinate spaces, like cartesian position of the end effector, by an estimation of the required joint angles with regard to collisions based on inverse kinematics. The commanded joint trajectories are processed by low-level controllers that unify model based forward signals and feedback signals for compensation the current error.

As increasingly complex robotic systems find their way into new application areas, the separation of human and robot work spaces is not feasible. Human-robot interaction (HRI), that aims for e.g. collaborative work or therapeutic use, makes high demands on control architectures and the robot structure. One requirement is a safe operation, since the robot interacts with a human user, whereas classical stiff actuators have a high potential of injury. As the risk analysis of head injuries on collision with robotic actuators by Zinn et al. [2004] shows, one way to lower the risk of injury is the reduction of the inertia of the moving parts of the robot. This led to the development of light-weight robots, a class of robotic manipulators that aim at mobility and safety in unknown environments. Light-weight robots reach a high payload to weight ratio and often integrate advanced sensor capabilities, detection of external collisions and gravity compensation for interaction with humans. A typical example is the 7-DOF Kuka-DLR light-weight arm [Hirzinger et al., 2002] with a weight of 14kg and 10kg of payload. But light-weight robot structures cause higher elasticity and pose further challenges on high frequency and precise control as well as vibration compensation [Albu-Schäffer et al., 2007]. A second option to enhance the safety of manipulators is to decrease the stiffness of the actuator. As compliance is the complementary concept of stiffness and terminology in literature is diverse, variable stiffness, adjustable compliance, variable compliance, adjustable stiffness or controllable stiffness are used to describe the flexibility of a robot. Implementations on robots are subdivided into systems with passive and active compliance. Active compliance refers to an actuator that mimics the compliant behavior of a spring by sophisticated control [Albu-Schäffer et al., 2011], but no energy storage or shock absorbance can be achieved as in the case of passive compliance, which elastically decouples the actuator from the load. Common examples of passive compliant actuators are pneumatically driven or incorporate Series Elastic Actuators (SEA) [Pratt and Williamson, 1995], a mechanism of a spring in series with a classical stiff actuator. Several compliant actuator concepts have been proposed that add elastic elements to the joints for enhanced safety and the aim for a reduced power consumption by temporarily storing energy in the joints [Ham et al., 2009]. Ranging from bio-inspired robot designs like a hexapod [Schneider et al., 2014] with elastomer coupled actuators [Paskarbeit et al., 2013], a quadruped robot with compliant legs based on a spring mechanism [Rutishauser et al., 2008] or a humanoid robot like the

COMAN that integrates spring-coupled actuators [Tsagarakis et al., 2011]. Further work aims at adjustable compliance for passive compliant actuators, like antagonistic-controlled stiffness by the use of pneumatic muscles [Tondu et al., 2005; Verrelst et al., 2006; Ikemoto et al., 2015; Büchler et al., 2016] or variation of the spring preload, material properties and transmission ratios as discussed by Wolf et al. [2016]. But control of compliant actuators is difficult and requires a model of the nonlinear properties of the elastic elements, which often prevents analytical modeling, underlies manufacturing tolerances and changes its physical properties dynamically or caused by wear-and-tear.

In recent years, an increasing number of soft robots have surfaced in various forms and fields. In their development, researchers have been driven by various motivations. Bioroboticists, for instance, refer to the "understanding by building" approach which is well established in order to complement experimental and theoretical work on biological mechanisms. Prominent examples include artificial salamanders [Ijspeert et al., 2005], hexapods [Schilling et al., 2013; Schneider et al., 2011], snakes [Transeth et al., 2008], worms [Seok et al., 2013], octopus [Calisti et al., 2011], or smaller quadrupeds [Spröwitz et al., 2013]. In a similar vein, researchers have build humanoid robots like [Marques et al., 2010], [Shirai et al., 2011], or [Ott et al., 2013] with the background motivation to understand the role of embodiment in cognition [Pfeifer and Bongard, 2006] and human-like motor behavior [Tsagarakis et al., 2009]. All the mentioned approaches share the explicit or implicit interest to investigate the interplay of morphology and computation, most prominently phrased under the notion of morphological computation [Hauser et al., 2011]. On the other hand, there is an increasing interest from several application fields in soft robotics. From the perspective of safe human-robot interaction, intrinsically safe and fully passive compliant soft-robot platforms like the Bionic Handling Assistant (BHA, [Grzesiak et al., 2011], see Table 2.1) or components like a soft skin [Duchaine et al., 2009] are being developed. These platforms share experimental mechanics and actuation designs, for which kinematic or dynamic models are hardly available and often have to be approximated, e.g. for the BHA [Rolf and Steil, 2012]. Typically, standard methods of model-based control cannot be applied easily. But also learning methods are not easily applicable. The main reason is that soft-mechanisms often involve high-dimensional actuation with heavy redundancy, have slow and complex mechanical dynamics that often include hysteresis, and exhibit long control delays. Problems for holistic learning approaches thus include that exploration suffers from the curse of dimensionality and simulations are not available again due to the lack of models. The generation of training examples from the robot itself is difficult and costly, because the mechanism has to be executed for each sample, and the reproducibility of actions and their results is limited. It was previously shown by Rolf and Steil [2014], that the kinematic control of such robots can be effectively improved by novel, biological inspired learning schemes that do not rely on exhaustive exploration. However, also lower levels of control pose significant challenges on soft robots, yet being essential to exploit the robots' full potential for safe physical human-robot interaction. While the robot's soft material and actuation permit close spatial proximity between human and robot without posing a threat to the human, its material properties cannot be productively harnessed of the shelf. It is desirable to be able to freely move the robot to configure its posture [Lemme et al., 2013] or teach in movements to be executed [Akgun et al., 2012]. Typical application scenarios are small scale production lines in which expert programming of the robot is an essential cost factor. In such scenarios, naive users should be able to "program" how a robot executes a task by kinesthetically teaching it. The use of such active compliant control modes have already been shown in industrial contexts [Wrede et al., 2013], but the very control so far required fast and accurate force sensing as well as accurate models of the robot itself, both of which are typically unavailable on soft robots.

Motivation From Biology:
Cerebral and Sub-Cortex Motor Control in Mammalians Concerning all previously mentioned challenges in motor control, the question arises how complex motion generation is realized in animals and humans. In particular, with reference to the high dimensionality of actuation and correlation between muscle fibers as well as further complex and compliant properties of the musculoskeletal system. Even with the assumption that the body optimized its structure under evolutionary pressure resulting in a simplification (i.e. linearization) of the control problem (known as morphological computation [Pfeifer and Gómez, 2009]) high-dimensional nonlinear relations between sensory input, abstract high-level goals and motor signals remain.

Although huge efforts have been made in understanding the motor system of the brain, even the functional role of primary motor cortex (M1) area is still controversially discussed. The Servo Hypothesis targets on understanding of low-level control by combination of distributed feed-forward models as proposed by Schweighofer et al. [1998]. Further, higher-level feed-forward estimates in combination with feedback loops are assumed to reach higher-level goals [Wolpert and Ghahramani, 2000]. To address the problem of the high dimensionality, the concept of synergies between muscles [An et al., 2014] assumes that complex motions can be generated by mixing basis functions of muscle activations. Further concepts include that complex motions are composed of simpler motion primitives [Mussa-Ivaldi and Bizzi, 2000]. Beside experiments that show a decerebrated cat performing several gait patterns [Whelan, 1996], recent research indicates that the motor cortex does even not play a crucial role in motion execution [Kawai et al., 2015]. Lower sub-cortical areas seem to be responsible for motion execution and the motor cortex performs modulation and learning. This view is also supported by Schieber [2000] with the statement that one of the tasks of the primary motor cortex includes the adaptation of motions to internal or external conditions. The discussion by Graziano [2015b] points out that besides the view of a homunculus-like map of muscles and a population coding of spatial muscle activations, a third view emerges in form of a represented action-map. This view is supported by the activation of specific basic actions in relation to stimulation of different M1 cortex regions, e.g. *hand-to-mouth* or *reach-*

to-grasp movements [Graziano, 2015a]. Additionally, [Scott, 2008] argues that high and low-level signals modulate activations in M1 and response patterns of single neuron populations are dependent on trajectory shapes as well as load situations of the actuated limbs. An action-map representation is addressed by Optimal Feedback Control [Scott, 2004], a proposed conceptual framework that tries to keep motion variability in cases where the task performance is not affected. As it can be seen from the previous discussion, the role of the primary motor cortex is not yet revealed but it can be assumed that the primary motor cortex consolidates multi-modal high-level information as well as low-level signals. Moreover, it is crucial for learning and the adaptation of movements to parameterizations of various abstraction levels, whether motion execution seems to be located in sub-cortical regions and gets modulated by the primary motor cortex.

Skill Learning for Robots:
Parameterized Skills Advanced robotic systems face non-static environmental conditions which require context-dependent adaptation of motor skills. Approaches that optimize motions for a given task by reinforcement learning, like object manipulation [Günter, 2009] or walking gait exploration [Cai and Jiang, 2013], deal only with a single instance of a potentially parameterized set of tasks. In many cases, a low-dimensional parameterization that covers the variance of a task exists. For example, consider reaching and grasping under various obstacle positions and object postures [Ude et al., 2007; Stulp et al., 2013], throwing of objects at parameterized target positions [Silva et al., 2014] or playing table tennis using motion primitives that are parameterized with respect to the current ball trajectory [Kober et al., 2012]. A full optimization for each new task parameterization from a reasonable initialization, which was acquired by e.g. kinesthetic teaching, means that many computations and trials need to be performed before the task can be executed. This impedes immediate task execution and is highly inefficient for executing repetitive tasks under some structured variance.

Recent work addresses this issue by introducing parameterized motor skills that estimate a mapping between the parameterization of a task and corresponding solutions in policy parameter space [Ude et al., 2007; Pastor et al., 2013; Stulp et al., 2013; Silva et al., 2014; Mülling et al., 2010; Kober et al., 2012; Matsubara et al., 2011; Reinhart and Steil, 2015; Baranes and Oudeyer, 2013]. Generation of training data for the update of such parameterized skills requires the collection of optimized policies for a number of task parameterizations. In previous work, each training sample is based on a full optimization for a new task parameterization starting from a fixed initialization [Silva et al., 2012, 2014], or gathered in demonstrations e.g. by kinesthetic teaching [Ude et al., 2007; Stulp et al., 2013; Matsubara et al., 2011; Reinhart and Steil, 2015]. On the one hand, requesting demonstrations from a human teacher for many task parameterizations is not only time-consuming, but also includes the risk of collecting very different solutions to similar tasks due to the redundancy of the problem. Solutions on a smooth manifold are a prerequisite to allow for generalization for unknown tasks by machine learning algorithms. On the other hand, full optimization from a single initial condition requires many rollouts and ignores the already acquired knowledge about the motor skill. A further method to encode the behavior of dynamical systems to generate trajectories in relation to a task parameterization are Task-Parameterized Gaussian Mixture Models (TP-GMM) [Calinon et al., 2013; Calinon, 2016]. Demonstrations are encoded as Gaussian Mixture Models in relation to multiple reference frames like via-points or start/end positions. Relative to each frame, Gaussian Mixture Model parameters that represent the demonstrations are estimated by an EM algorithm. Generation is based on the joint distribution of all Gaussian mixture models.

2.2 A Novel Conceptual Framework for Parameterized Skill Learning

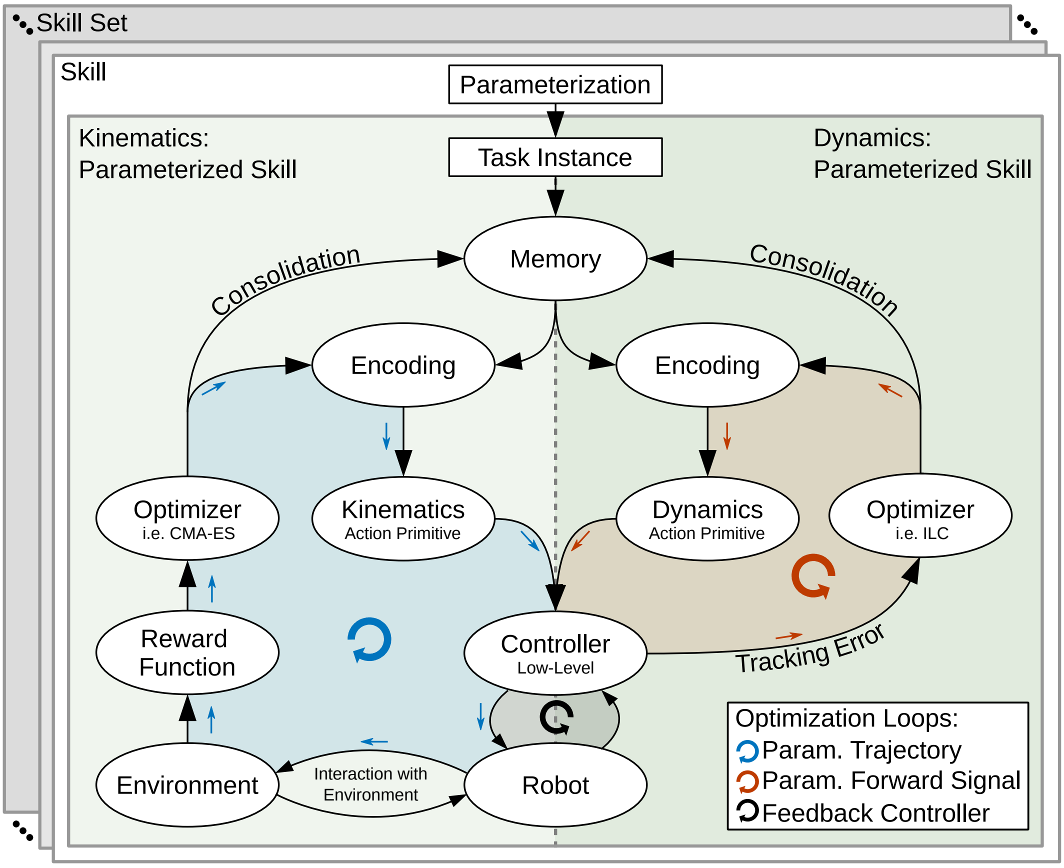


Figure 2.1: System diagram of the proposed Skill Learning Architecture. Successful task execution in real world scenarios is composed out of a kinematics (left) and dynamics (right) representation of actions.

This work will refer to the term skill learning in the context of robot action generation. A skill is the ability of the robot to carry out a task with a determined result. In comparison to classical robotic applications, it is assumed that the task is not static and is affected by some structural perturbation. Task variability could e.g. include variable positions of obstacles, goal position and orientations, variable weight of manipulated objects or a variable duration of the action. For each execution, the robot has to adapt its movements according to the parameterization of the current task instance it has to face. It is assumed that a high-level parameterization is available that describes the full variability for a given task. The remainder of this work, will refer to a *Parameterized Skill* (PS) as a memory that performs the generalization from a continuous task parameterization, that defines the current task instance, to a parameterization that generates an appropriate movement of the robot. The parameterized skill is trained with successful examples of movements for the current task parameterization (task instance). Fulfillment of the task can be measured in terms of a threshold on an objective function, like an estimation of a reward for the quality of an executed movement. The representation of movements is divided into a kinematics and dynamics representation of the skill. The kinematics representation results in required joint angle trajectories that have to be executed to fulfill a given task instance. Complex dynamics of the robot and interaction forces that may occur for successful task completion can prevent the precise execution of the required actions of the robot. The dynamics representation of the skill generalizes forward signals that support the low-level controller to perform a precise execution of the estimated joint angle trajectories by a representation of the dynamics of the robot and its interaction in relation of the task parameterization. As an example, consider the task of opening a door with a highly compliant robot. The system may have learned that a handle has to be rotated in relation to its attachment point on the door as joint angle representation (kinematics). But the rotation cannot be executed accurately by the highly compliant robot, since the handle includes a spring mechanism that works against the action performed by the robot. The dynamic representation covers the unmodeled dynamic properties of the interaction and generates a force that compensates for the spring mechanism of the door handle. A structural overview of the conceptual framework is shown in Figure 2.1. For a specific situation, one skill from a set of skills is selected. The current task instance is defined by a parameterization of the selected skill. A memory structure maps the task parameterization to an action representation. The action representation is encoded into a kinematics and dynamics representation of the current task. The resulting control signals are forwarded to a low-level controller that generates movements of the robot system. The robot system interacts with the environment and each action is assessed by a reward function. Based on the reward it is decided if the current action fulfilled the requirements of the given task. As indicated by the arrow symbols, multiple optimization loops are responsible for skill learning. For each task instance, an optimization of the kinematics (blue, Chapter 3-4) and dynamics (red, Chapter 6) representation is performed. Additionally, each primitive is executed by the low-level controller (black, Chapter 5). Further feedback occurs during primitive execution due to the interaction with the environment. This process has to be repeated for multiple task instances as well as multiple skills that form a task set. Note, this work is restricted to one parameterized skill and will not elaborate skill sets.

To gather training data for the parameterized skill, the system has to optimize the kinematics representation of a skill by maximization of the estimated reward of an executed movement, given the current task instance. The dynamics representation is optimized simultaneously in relation to the commanded joint angle trajectory, where the goal is the reduction of the tracking error of the low-level controller.

Formalization Action generation is performed by policies $\pi_{\boldsymbol{\theta}}$ that are parameterized by $\boldsymbol{\theta} \in \mathbb{R}^F$. Further, it is assumed that tasks are parameterized by $\boldsymbol{\tau} \in \mathbb{R}^E$ with $E << F$. Task instances defined by $\boldsymbol{\tau}$ are distributed according to the probability density function $P(\boldsymbol{\tau})$. The task parameterization $\boldsymbol{\tau}$ reflects the variability of the task, e.g. position of obstacles, target positions or loads attached to the end effector. With reference to [Silva et al., 2014], this thesis introduces the notion of a parameterized skill, which is given by the function $\mathrm{PS} : \mathbb{R}^E \rightarrow \mathbb{R}^F$, that maps task parameters $\boldsymbol{\tau}$ to a policy parameterization $\boldsymbol{\theta}$. The goal is to find a parameterized skill $\mathrm{PS}(\boldsymbol{\tau})$ that maximizes $\int P(\boldsymbol{\tau}) J(\pi_{\mathrm{PS}(\boldsymbol{\tau})}, \boldsymbol{\tau}) d\boldsymbol{\tau}$, with $J(\pi, \boldsymbol{\tau}) = \mathbb{E}\left\{R(\pi_{\boldsymbol{\theta}}, \boldsymbol{\tau}) | \pi, \boldsymbol{\tau}\right\}$ as the expected reward for using policy $\pi_{\boldsymbol{\theta}}$ to solve a task $\boldsymbol{\tau}$. The reward function $R(\pi_{\boldsymbol{\theta}}, \boldsymbol{\tau})$ assesses each action of the robot defined by the policy $\pi_{\boldsymbol{\theta}}$ with respect to the current task parameterization $\boldsymbol{\tau}$. In case of a representation of the kinematics, the parameterization $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathrm{K}}$ of policy $\boldsymbol{Q} = \pi_{\boldsymbol{\theta}} \in \mathcal{R}^{N_\pi x T}$ represents trajectories in joint angle $(N_\pi = N_{\mathrm{dof}})$ or end effector $(N_\pi = 3)$ space. In case of an additional representation of the dynamics of a task, the parametrization $\boldsymbol{\theta} = [\boldsymbol{\theta}_{\mathrm{K}}, \boldsymbol{\theta}_{\mathrm{D}}]$ of the policy represents further forward signals encoded as $\boldsymbol{\theta}_{\mathrm{D}}$. The resulting policy $\begin{pmatrix} \boldsymbol{Q} & \boldsymbol{U}^{\mathrm{FFWD}} \end{pmatrix} = \pi_{\boldsymbol{\theta}}$ provides a trajectory representation $q_{t,j}$ as well as forward signal that support the feedback controller $u_{t,j}^{\mathrm{FFWD}}$ for a primitive at time $t = 1 \ldots T$ and joint $j = 1 \ldots N_{\mathrm{dof}}$.

2.2.1 Key Aspects of the Contribution of this work in Relation to Previous Work

As discussed in the previous section, skill learning on real robotic systems that interact with humans and the environment is a challenging problem. Skill learning that is based on motion primitives in relation to a high-level task parameterization was demonstrated as a solution to overcome the challenges of high-dimensional state spaces in previous work. Impressive tasks could be tackled with these approaches, like dart throwing [Silva et al., 2012] or object transport [Stulp et al., 2013]. But nevertheless, those works do not tackle learning of dynamic properties, perfect execution of the motions on the robot system is assumed.

In this work, an architecture for skill learning is proposed as outlined in Figure 2.1. A parameterized memory is responsible for generalization of robotic actions for a given task instance defined by the task parameterization. In comparison to previously proposed skill learning architectures, the memory generalization results in two distinct modalities:

- Representation of the kinematics of the skill (Figure 2.1, left side)
- Representation of the dynamics of the skill (Figure 2.1, right side)

Complex dynamics of the robot or forces that occur during interaction with the environment, e.g. obstacle manipulation, can impede the low-level controller and prevent successful task execution. Therefore, an additional representation of the dynamics is proposed. In comparison to classical robot control methods, a forward signal to support the low-level controller is generalized based on a high-level task parameterization.

In comparison to existing methods, the work presented in this thesis does not rely on offline methods or slow gradient descent and is able to deal with incremental consolidation of new samples. One of the most crucial competences of a system for online learning is the ability to quickly adapt to new tasks or extend the current skill representation. Learning parameterized skills from human demonstrations or multiple executions of stochastic optimization is costly as it is time consuming. For this reason, this work provides a framework that allows an integration of state-of-the-art optimization algorithms for policy search, i.e. by CMA-ES, instead of optimizing meta-parameters of policies [Kober et al., 2012] and does not rely on library based approaches, as in [Mülling et al., 2010].

The first option to allow for an efficient skill learning is the reduction of the number of required training samples. This work investigates an incremental algorithm to establish parameterized skills, that reuses previous experience to successively improve the optimization process [Queißer et al., 2016]. In contrast to [Silva et al., 2012, 2014], the optimizer is initialized with the current estimate of the iteratively trained skill. Further, a cost term is proposed and used as an additional objective for optimization of the kinematics representation of the skill. An analysis on toy data demonstrates improved generalization capabilities due to the selection of solutions that lead to a beneficial skill representation of the parameterized skill.

A further option to speed up the optimization is the reduction of the dimensionality of the search space. In comparison to previous work, it is proposed to rely to the space of task parameterizations for a reduction of the dimensionality. The parameterized skill performs a mapping from the low-dimensional space of task parameterizations to the high-dimensional space of the action parameterization. The proposed optimization in hybrid spaces allows for a fast coarse search in the low-dimensional input space of the parameterized skill and a refinement of the actions by a search in the full parameterization of the motions.

Relation to Inverse Model Learning In comparison to the exploration of mappings, for e.g. inverse kinematic models like by the Goal Babbling algorithm proposed by Rolf et al. [2010] or its extension to skill representations by Reinhart [2017], the exploration of a parameterized skills is not able to explore a task parameterization for arbitrary policy parameterizations. Therefore the mechanisms for learning such a memory cannot be transferred and used in the same way for learning of parameterized skills. Goal Babbling for example, relies on the ability to examine the current quality of the mapping of a smoothly moving parameterization in task space. This is not applicable for scenarios that are tackled in this thesis in which the task parameterization is typically given by environmental conditions and cannot be influenced by the learning method.

Relation to Deep Reinforcement Learning Approaches Recently, approaches that are based on *Deep Learning* [1] gained attention in the robot control community. These architectures focus on the processing of raw sensory signals, since the deep learning architectures are able to extract low-dimensional features from high-dimensional input in an unsupervised manner. This work does not aim for the extraction of features, as it is assumed that reasonable low-dimensional features are already available. Furthermore, it is assumed that only a small amount of training samples can be gathered for exploration of the parameterized skill. Nevertheless, deep learning architectures could be used in synergy with the work of this thesis to perform an unsupervised extraction of low-dimensional features for the task parameterization of the proposed skill learning methods.

2.2.2 Component of a Skill Learning Architecture

This section will give an overview of the components of the proposed system architecture, as shown in Figure 2.1. The functional building blocks: memory, encoding and optimization will be discussed.

Memories

As introduced in Section 2.2, the memory component is given by the mapping function $\boldsymbol{\theta} = \mathrm{PS}(\boldsymbol{\tau})$ of the parameterized skill. For implementation of PS, nonlinear regression or associative representations can be considered. In comparison to nonlinear regression methods, associative memories have the benefit of completion of incomplete feature representations and bidirectional estimations, which is relevant for the proposed hybrid optimization in Section 4.2. A comprehensive review of current methods for nonlinear regression can be found in [Stulp and Sigaud, 2015]: a classification of regression models into function representations based on a weighted sum of basis functions or a mixture of linear models is presented. The authors argue that the representations that are based on the weighted sum of basis functions are a special case of the representations that are based on mixtures of linear models.

[1] It is referred to the term *Deep Learning* as stacked Restricted Boltzmann Machines (RBM) [Salakhutdinov and Hinton, 2009], convolutional networks [Wersing and Körner, 2003], Slow Feature Extraction (SFE) based architectures [Franzius et al., 2007] and further stacked networks, e.g. [Deng and Yu, 2014; LeCun et al., 2015], that transform signals from local to global and optionally from a fast to a slow context in multiple layers $n > 2$.

Therefore, a model is presented that unifies the representation of common learning methods as, among others, Locally Weighted Regression (LWR), Gaussian Mixture Regression (GMR), Radial Basis Function Networks (RBFNs), Gaussian Process Regression (GPR), Support Vector Regression (SVR), Extreme Learning Machine (ELM) or Backpropagation.

In the case of the parameterized skill, it can be assumed that only a low number of training samples are available. Each training sample has to be gathered by kinesthetic teaching or policy optimization which is costly since it requires interaction with the robot or repetitive executions of actions of the robot.

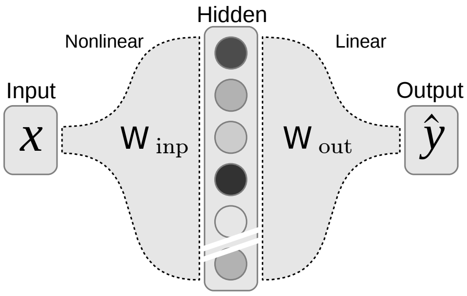


Figure 2.2: Structure of the ELM as function approximator. The input extension to the hidden layer is based on randomly selected input weights $\mathbf{W}_{\mathrm{inp}}$. The readout weights $\mathbf{W}_{\mathrm{out}}$ are estimated by means of linear regression.

This thesis refers to a single-layer feed-forward network with a random projection into the hidden layer and a linear readout, as known in literature as Randomized Neural Network (RNN) [Schmidt et al., 1992] in case of a linear regression on the random projection including a bias, Random Vector Functional Link (RVFL) [Pao et al., 1994] in case of a linear regression on the random projection and the untransformed input pattern, and as Extreme Learning Machine (ELM) that performs linear regression only on the random projection of the input pattern. Literature shows that these methods achieve a competitive performance in comparison to other state-of-the-art nonlinear regression methods [Liu et al., 2012; Enache and Dogaru, 2015]. Further, estimation of the parameterization does not require a slow gradient decent because the linear readout and hyper-parameters are easy to tune for real world applications.

Since all three variants have huge similarities, the discussion of the methods will be restricted to ELMs. The parameterized skill implemented as ELM is defined as

$$\mathrm{PS}_i(\boldsymbol{\tau}) = ELM(\boldsymbol{\tau}) = \sum_{j=1}^{N_{\mathrm{H}}} \mathbf{W}_{ij}^{out} \boldsymbol{h}_j(\boldsymbol{\tau}) \;\; \forall i = 1, ..., F, \tag{2.1}$$

with N_{H} hidden nodes and output dimensionality F. The hidden activation $\boldsymbol{h}$ is defined as

$$\boldsymbol{h}_j(\boldsymbol{\tau}) = \sigma\left(\sum_{k=1}^{E} \mathbf{W}^{inp}_{j,k}\boldsymbol{\tau}_k + \boldsymbol{b}_j\right) \quad \forall j = 1, ..., N_{\mathrm{H}} \tag{2.2}$$

with input dimensionality E. The nonlinearity of hidden states is introduced by sigmoid activation function $\sigma(x) = (1 + e^{-\alpha x})^{-1})$ with slope parameter α. In comparison to methods based on vector quantization, the selection of a random projection simplifies model selection and does not require an adaptation of prototypes. For training, it is assumed that H is the collection of all N_{H} hidden states for all N_{tr} samples of a dataset,

$$\mathbf{H} = \begin{bmatrix} h(\boldsymbol{\tau}_1) \\ \vdots \\ h(\boldsymbol{\tau}_{N_{\mathrm{tr}}}) \end{bmatrix} = \begin{bmatrix} h_1(\boldsymbol{\tau}_1) & h_2(\boldsymbol{\tau}_1) & \cdots & h_{N_{\mathrm{H}}}(\boldsymbol{\tau}_1) \\ h_1(\boldsymbol{\tau}_2) & h_2(\boldsymbol{\tau}_2) & \cdots & h_{N_{\mathrm{H}}}(\boldsymbol{\tau}_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\boldsymbol{\tau}_{N_{\mathrm{tr}}}) & h_2(\boldsymbol{\tau}_{N_{\mathrm{tr}}}) & \cdots & h_{N_{\mathrm{H}}}(\boldsymbol{\tau}_{N_{\mathrm{tr}}}) \end{bmatrix}. \tag{2.3}$$

This allows the definition of the parameterized skill in matrix notation as $\mathrm{PS}(\boldsymbol{\tau}) = \mathbf{H}\mathbf{W}_{out}$. Learning is performed by minimization of the error between the output and desired targets $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_{N_{\mathrm{tr}}}]^\top$, given by

$$\underset{\widehat{\mathbf{W}}_{\mathrm{out}}}{\operatorname{argmin}} \, ||\mathbf{H}\mathbf{W}_{\mathrm{out}} - \boldsymbol{\Theta}||. \tag{2.4}$$

As the parameterization of the learner can be estimated by linear regression, implementation of the learner can be realized by several well established methods as discussed in the following:

a) Recursive Least Squares (RLS):

One prominent method solving linear least squares problems is Recursive Least Squares (RLS). RLS is able to process sequentially available training data for an update of $\widehat{\mathbf{W}}_{\mathrm{out}}$ under consideration of an optional exponential forgetting of old training samples. Those properties make RLS an interesting candidate for the implementation of the parameterized skill. An ELM variant based on sequential learning is presented in [Liang et al., 2006]. The incremental update of the readout weights is given by

$$\widehat{\mathbf{W}}_{\mathrm{out}}(k+1) = \widehat{\mathbf{W}}_{\mathrm{out}}(k) + \underbrace{\frac{\mathbf{P}(k) \cdot \boldsymbol{h}(\boldsymbol{\tau}_{k+1})}{\lambda + \boldsymbol{h}(\boldsymbol{\tau}_{k+1})^\top \cdot \mathbf{P}(k) \cdot \boldsymbol{h}(\boldsymbol{\tau}_{k+1})}}_{\text{Kalman Filter Gain}} \cdot \underbrace{\Big(\boldsymbol{\theta}_{k+1} - \boldsymbol{h}(\boldsymbol{\tau}_{k+1})^\top \cdot \widehat{\mathbf{W}}_{\mathrm{out}}(k)\Big)}_{\text{Innovations}}, \tag{2.5}$$

with

$$\mathbf{P}(k+1) = \frac{1}{\lambda} \cdot \Big(\mathbf{P}(k) - \gamma(k) \cdot \boldsymbol{h}(\boldsymbol{\tau}_{k+1})^\top \cdot \mathbf{P}(k)\Big). \tag{2.6}$$

Exponential forgetting is given by $0 < \lambda \leq 1$. For $\lambda = 1$ the update results in RLS without exponential forgetting of old training samples.

b) **Regularized Least Squares:**
A further prominent method for solving linear regression problems is Regularized Least Squares. It adds a further minimization constraint regarding the readout weights, the resulting error is given by

$$||H\mathbf{W}_{\text{out}} - \mathbf{\Theta}||^2 + \gamma||\mathbf{W}_{\text{out}}||^2. \tag{2.7}$$

Usually, the L_2-norm is used due to its closed form solution that is called ridge regression or Tikhonov regularization [Tichonov and Arsenin, 1977]. The solution of the optimization problem of $\widehat{\boldsymbol{w}}_{\text{out}}$ in case of ridge regression is given by

$$\widehat{\mathbf{W}}_{\text{out}} = (\mathbf{H}^\top\mathbf{H} + \gamma\mathbf{I})^{-1}\mathbf{H}^\top\mathbf{\Theta}. \tag{2.8}$$

A ELM variant that incorporates regularization is presented in [Deng et al., 2009; Neumann and Steil, 2013]. Further work of [Huynh and Won, 2011] introduces a combination of sequential and regularized learning. Resulting in an incrementally updated estimation of the readout weights for sequential data chunks,

$$\widehat{\mathbf{W}}_{\text{out}}(k) = \widehat{\mathbf{W}}_{\text{out}}(k-1) + \mathbf{L}^{-1}(k)\mathbf{H}^\top(k)(\mathbf{T}(k) - \mathbf{H}(k)\widehat{\mathbf{W}}_{\text{out}}(k-1)), \tag{2.9}$$

with

$$\mathbf{L}(k) = \mathbf{L}(k-1) + \mathbf{H}(k)^\top\mathbf{H}(k). \tag{2.10}$$

The initialization is given by

$$\widehat{\mathbf{W}}_{\text{out}}(0) = \mathbf{L}^{-1}(0)\mathbf{H}^\top(0)\mathbf{T}(0), \qquad \textit{with } \mathbf{L}(0) = \mathbf{H}^\top(0)\mathbf{H}(0) + \lambda\mathbf{I}. \tag{2.11}$$

An additional weighting of the training set can be performed to modulate the importance of each presented training sample or be applied as Iteratively Reweighted Least Squares (IRLS) to approximate least square problems regularized by L_1-norms or even non-convex fractional norms [Aggarwal et al., 2001; Chartrand and Yin, 2008].

c) **Bayesian Linear Regression:**
In addition to the previously presented approaches, Bayes Linear Regression allows the estimation of a posterior distribution of the readout weights, e.g. [Bishop, 2006; Soria-Olivas et al., 2011], defined for output i as

$$p(\widehat{\boldsymbol{w}}_{\text{out},i}|\mathbf{\Theta}) = N(\widehat{\boldsymbol{w}}_{\text{out},i}|\boldsymbol{m}_{N,i}, \mathbf{S}_N). \tag{2.12}$$

The prior $\mathbf{S}_0 = \alpha^{-1}\mathbf{I}$, is assumed to be zero mean and isotropic. Posterior distribution over $\widehat{\boldsymbol{w}}_{\text{out},i}$ is given by $\boldsymbol{m}_{N,i} = \beta\mathbf{S}_N\mathbf{H}^\top\mathbf{\Theta}_{i,*}$ and $\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\mathbf{H}^\top\mathbf{H}$. Parameter $\beta = 1/\sigma_{\text{tr}}^2$ is given by the inverse of the variance of the training data.

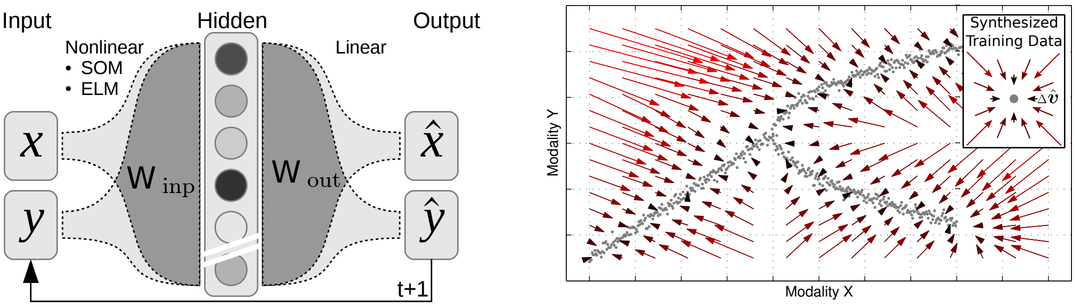
The final output of the parameterized skill is given by the predictive distribution, as

$$p(PS_i|\boldsymbol{\tau}, \alpha, \beta) = \mathcal{N}(PS_i|\widehat{\mathbf{W}}_{\text{out},i}^{\top}\boldsymbol{h}(\boldsymbol{\tau}), \sigma_N^2(\boldsymbol{\tau})), \tag{2.13}$$

with variance

$$\sigma_N^2(\boldsymbol{\tau}) = \frac{1}{\beta} + \boldsymbol{h}(\boldsymbol{\tau})^{\top}\mathbf{S}_N\boldsymbol{h}(\boldsymbol{\tau}). \tag{2.14}$$

Associative Memories



(a) (b)

Figure 2.3: Associative network structure. Feedback of the output results in a dynamic behavior that is visualized as vector field (b).

Associative networks have been motivated as biologically inspired learning methods. One basic concept of networks based on feedback connections, is to employ an auto-associative network that minimizes an energy function or follows a gradient to reach a local minimum that represents the distribution of the training data. For association of different modalities, the state description of all M modalities are concatenated into $\boldsymbol{v} = [\boldsymbol{v}^{(1)^{\top}} \cdots \boldsymbol{v}^{(M)^{\top}}]^{\top}$ and used as target for the auto-encoder. This results, in case of the parameterized skill, in the association of $\boldsymbol{v}^{(1)} = \boldsymbol{\tau}$ and $\boldsymbol{v}^{(2)} = \boldsymbol{\theta}$ in $\boldsymbol{v} = [\boldsymbol{v}^{(1)}\, \boldsymbol{v}^{(2)}]^{\top}$. Network estimates $\hat{\boldsymbol{v}}_t$ and network dynamics is induced by assignment $\boldsymbol{v}_{t+1} \leftarrow \hat{\boldsymbol{v}}_t$. By fixation of single modalities or even dimensions, it is possible to query the memory for a given (incomplete) pattern. Variations of the initial state of the network allow for selection of solutions in case of ambiguous data (multiple attractors). In the following an overview over existing techniques for associative memories will be given, further models, e.g. prototype based, can be found in [Reinhart, 2011].

a) Hopfield Networks:

Hopfield networks are associative networks based on biologically motivated Hebbian learning [Hopfield, 1982], for binary pattern vectors. Later extensions to logistic functions allow the representation of graded responses [Hopfield, 1984]. The iterative update of the activation of the network is given by

$$v_i \leftarrow \begin{cases} +1 & \text{if } \sum_j w_{ij} v_j \geq \alpha_i, \\ -1 & \text{otherwise.} \end{cases} \tag{2.15}$$

with thresholds α_i The respective energy function of the network is defined as

$$E = -\frac{1}{2} \sum_{i,j} w_{ij} v_i v_j + \sum_i \alpha_i v_i \tag{2.16}$$

and is minimized by every update step towards a local minimum. Training can be performed by the Hebbian learning rule $w_{ij} = \frac{1}{N_{\text{tr}}} \sum_{\mu=1}^{N_{\text{tr}}} \epsilon i^{\mu} \epsilon_j^{\mu}$, for all training patterns $\boldsymbol{\epsilon}$. Training patterns are represented by a local minima of the energy function, but following the update rule, Equation 2.15, can result in local minima that do not present training data as well as spurious patterns.

b) **Restricted Boltzmann Machines (RBM):**
Restricted Boltzmann Machines (RBMs) [Smolensky, 1986; Freund and Haussler, 1992] can be interpreted as an extension of Hopfield Networks. They are extended by a probabilistic state description and a separation of a visible and a hidden layer. RBMs gained attention for successful application in classification tasks in hierarchical configurations. For real valued visible layers a Gaussian-Bernoulli-RBM can be considered [Hinton and Salakhutdinov, 2006; Cho et al., 2013]. The application of logistic functions allow for representation of the visible layer in continuous space. By iterative estimation of the hidden layer based on the the visible layer and vise versa, a completion/association can be carried out. As for the Hopfield network, iterative updates of the visible layer minimize the energy of the network. In case of Gaussian visible nodes, the energy is defined as

$$E(v, h) = \frac{||\boldsymbol{v} - \boldsymbol{b}^v||^2}{2\sigma^2} - (\boldsymbol{b}^h)^{\top} h - \frac{\boldsymbol{v}^{\top} \mathbf{W} \boldsymbol{h}}{2\sigma^2}. \tag{2.17}$$

The activation of the binary hidden nodes in relation to the visible nodes is expressed as

$$P(h_j = 1 | \boldsymbol{v}) = \sigma \left(b_j^h + \frac{\boldsymbol{v}^{\top} \mathbf{w}_{*,j}}{\sigma^2} \right) \tag{2.18}$$

and for back-projection the activation of the visible nodes given the state of the hidden nodes is given by

$$P(v_i | h) = N(b_i^v + \mathbf{w}_{i,*}^{\top} \boldsymbol{h}, \sigma^2). \tag{2.19}$$

The bias of the hidden layer is denoted as $\boldsymbol{b}^h$ and for the visible layer as $\boldsymbol{b}^v$. Binary hidden activation is notated as $\boldsymbol{h}$ and visible layer as $\boldsymbol{v}$. Notation $\mathbf{w}_{*,j}$ indicates the selection of one vector of the matrix $\mathbf{W}$. Training of the weights of the network is discussed in [Hinton, 2002]. The Boltzman Machines are not designed for a continuous data representation due to the binary state descriptions. Additionally, training usually requires large amounts of training data and is sensitive to parameter selection, in particular for Gaussian nodes.

c) **AELM and ARBF as Parameterized Skill Memories (PSM):** The associative learning introduced as Parameterized Skill Memory (PSM), is based on an auto-encoder that is implemented by programming of a multiple stable attractor dynamics. It is assumed that an induced error of the estimate of the auto-encoder generates a $\Delta\hat{\boldsymbol{v}}_t = \hat{\boldsymbol{v}}_t - \boldsymbol{v}_t$ that moves the next state of the network $\boldsymbol{v}_{t+1} \leftarrow \boldsymbol{v}_t + \Delta\hat{\boldsymbol{v}}_t$ closer to the distribution of the training data, as illustrated in Figure 2.3a. One implementation of the model is based on an explicit encoding of a vector field, as shown in Figure 2.3b. It was introduced as Associative Extreme Learning Machine (AELM) [Reinhart and Steil, 2011; Reinhart, 2011] as it incorporates a random, non-recurrent, and nonlinear projection into the hidden layer similar to the ELM. It is defined by

$$\boldsymbol{h}_t = \sigma(\mathbf{W}_{inp}\boldsymbol{v}_t). \tag{2.20}$$

The estimation of the output $\hat{\boldsymbol{v}}$ is based on a linear readout

$$\hat{\boldsymbol{v}}_t = \mathbf{W}_{out}\mathbf{h}_t. \tag{2.21}$$

Further work investigated associative reservoir computing including recurrent connections [Reinhart and Steil, 2011; Emmerich et al., 2013] . Training is performed by linear regression of $\mathbf{W}_{out}$, stable attractor points are imprinted by generation of synthesized sequences that point towards the training data, as shown in Figure 2.3b. But, convergence to the training distribution is not guaranteed, e.g. over-fitting of the learner can lead to poor solutions and an exponential number of generated training samples is required in relation to the dimensionality of the input.

A further implementation of this class of associative memories is based on a vector quantization approach. The Associative Radial Basis Function Network ARBF [Reinhart and Steil, 2012, 2014] is an associative learner based on hidden radial basis function nodes. Due to the radial basis functions, a stable attractor dynamics emerges as demonstrated in [Reinhart and Steil, 2012]. For this case, the hidden layer is estimated by

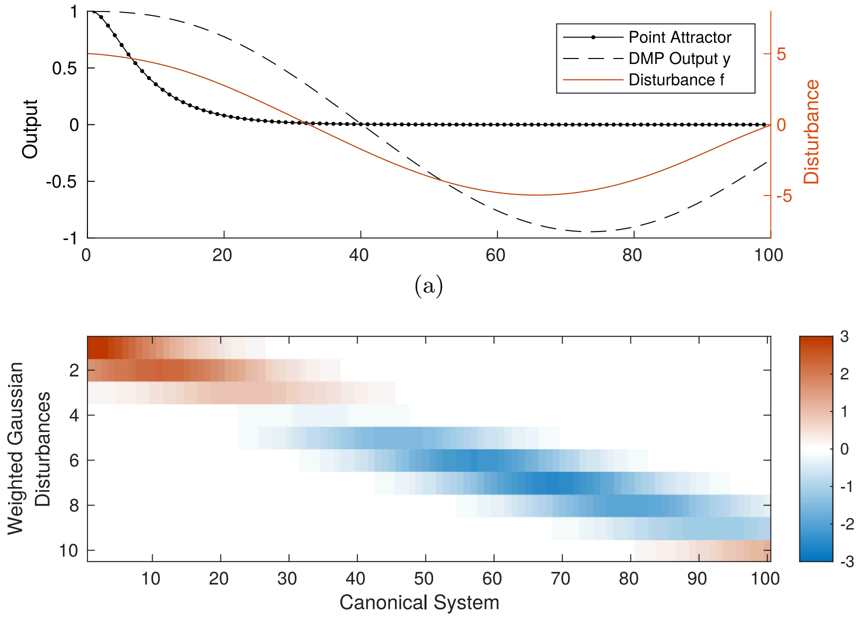
$$h_i(\boldsymbol{v}) = \frac{exp(-\sum_{m=1}^{M}\beta^{(m)}||\boldsymbol{v}^{(m)} - \boldsymbol{c}_i^{(m)}||^2)}{\sum_{j=1}^{N_\mathrm{H}} exp(-\sum_{m=1}^{M}\beta^{(m)}||\boldsymbol{v}^{(m)} - \boldsymbol{c}_j^{(m)}||^2)}. \tag{2.22}$$

Balancing of the modalities, e.g. to keep equal influence for modalities with different a dimensionality, can be implemented by scaling factors $\beta^{(m)}$. Output mapping and iterative update are performed in a same way as in Equation 2.20 and Equation 2.21.

Trajectory Representation

The task parameterization, as well as the policy parameterization, are time invariant representations of movements. The encoding aims at an efficient representation of the temporal joint trajectories as well as control signals. This includes compression of the parameter space, noise suppression and good generalization capabilities. Besides simple encodings based on polynomial functions or splines, e.g. [Andersson, 1989; Hwang et al., 2003], most prominent methods for robotic trajectory generations are based on a nonlinear dynamical systems approach. Often, those dynamical systems incorporate a linear dynamical system that predominates a nonlinear modulation. A phase variable represents an internal clock and performs a smooth transition between nonlinear and linear dynamics to ensure stability at the end of the motion.

a) Dynamic Motion Primitives (DMP):



(a)

(b)

Figure 2.4: Illustrative example of a DMP based trajectory representation.

Dynamical systems for trajectory planning and control have been proposed as Dynamic Movement Primitives (DMP) in [Schaal, 2006], they have been widely used in different applications for robot control. The basic idea of DMPs is to modulate a movement produced by a stable second-order dynamical system that is perturbed by a complex nonlinear force term. The force term itself consists of a weighted sum of multiple predefined activation functions. However, Calinon et al. [2012] extended the DMP framework to a probabilistic formulation in which a simple attractor point is obtained for every single data point. It refers to a similar dynamical spring-damper system without considering the force term. Instead of estimating a force term, the trajectory of virtual attractor points are encoded with statistical tools such as Gaussian Mixture Models in the form of a joint probability distribution. The resulting planning scheme benefits from multiple advantages of dynamical systems e.g., robustness when facing perturbations and control over the compliancy of the task execution by tuning the tracking gains. It also takes advantage of automatic organization of basis activation functions. In [Malekzadeh et al., 2014a], the idea of trajectory attractors is extended to surface attractors using spatio-temporal dynamical systems.

This thesis refers to Dynamic Motion Primitives (DMP, [Schaal, 2006; Ijspeert et al., 2013]) for encoding of trajectories, because they are widely used in the field of motion generation. DMPs for point-to-point motions are based on a dynamical point attractor system

$$\ddot{y} = k_S(g - y) - k_D\dot{y} + f_{\text{DMP}}(x, \boldsymbol{\theta}), \tag{2.23}$$

that defines the output trajectory as well as velocity and acceleration profiles. The canonical system is typically defined as $\dot{x} = -\alpha x$ or as a linear decay $\dot{x} = -\alpha$ as in [Kulvicius et al., 2012] and limited to non-negative values. The shape of the primitive is defined by

$$f_{\text{DMP}}(x, \boldsymbol{\theta}) = \frac{\sum_{k=1}^{K} \exp(-\boldsymbol{V}_k(x - \boldsymbol{C}_k))\boldsymbol{\theta}_k}{\sum_{k=1}^{K} \exp(-\boldsymbol{V}_k(x - \boldsymbol{C}_k))}, \tag{2.24}$$

where a mixture of K Gaussians is used. $\boldsymbol{C}_k$ are the Gaussian centers and $\boldsymbol{V}_k$ define the variance of the Gaussians. The DMP is parameterized by the mixing coefficients $\boldsymbol{\theta}_k$. Efficient encoding of trajectories by weights $\boldsymbol{\theta}_k$ can be achieved by linear regression, as the output of the disturbance function f_{DMP} depends linearly on the weights. Fixed variances $\boldsymbol{V}_k$ and a fixed distribution of centers $\boldsymbol{C}_k$ are assumed as in [Reinhart and Steil, 2015]. Figure 2.4 shows an exemplary configuration of a DMP. Figure 2.4a shows the response of the point attractor ($\boldsymbol{\theta} = 0$), the response of the disturbance term f_{DMP} and the resulting output of the DMP. Figure 2.4b visualizes the weighted Gaussian disturbance terms $\exp(-\boldsymbol{V}_k(x - \boldsymbol{C}_k))\boldsymbol{\theta}_k$.

b) Gaussian Mixture Models (GMM):

Gaussian Mixture Regression (GMR) [Günter et al., 2007] shares a joint representation of the input and outputs in variable $\boldsymbol{u} = [t\ \boldsymbol{q}_t]^\top$, or in case of a dynamical system as $\boldsymbol{u} = [t\ \dot{\boldsymbol{q}}_t]^\top$. The relation of input, i.e. time, and output is modeled as probability density function

$$p(\boldsymbol{u}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.25}$$

with weighting factors π_k of a mixture of K Gaussian distributions with means $\boldsymbol{\mu}_k$ and covariances $\boldsymbol{\Sigma}_k$. The parameterization can be estimated by the Expectation Maximization algorithm (EM), as demonstrated in [Ghahramani and Jordan, 1994].

The recovering of the expected output from the conditional probability function is performed by

$$q(t) = \sum_{k=1}^{K} \frac{\pi_k \mathcal{N}(\boldsymbol{\mu}_{k,t}, \boldsymbol{\Sigma}_{k,t})}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{\mu}_{j,t}, \boldsymbol{\Sigma}_{j,t})} \left(\mu_{k,q} + \boldsymbol{\Sigma}_{k,qt} \boldsymbol{\Sigma}_{k,t}^{-1} (t - \mu_{k,t}) \right). \tag{2.26}$$

Where $\mu_{k,t}$, $\mu_{k,q}$, $\boldsymbol{\Sigma}_{k,t}$, $\boldsymbol{\Sigma}_{k,q}$, $\boldsymbol{\Sigma}_{k,tq}$ and $\boldsymbol{\Sigma}_{k,qt}$ are estimated by the separation of the covariance matrix and the means of the joint representation into the respective input and output components as shown in [Günter et al., 2007]. Due to the probabilistic representation, the variance of the conditional distribution can be recovered as well.

c) **Neurally Imprinted Vector Fields:**
The idea of Neural Imprinted Vector Fields [Lemme et al., 2014] follows the motivation of a vector field representation of a dynamical system, similar to the associative memory AELM as discussed before on page 30. The temporal dynamics of the trajectory is represented as

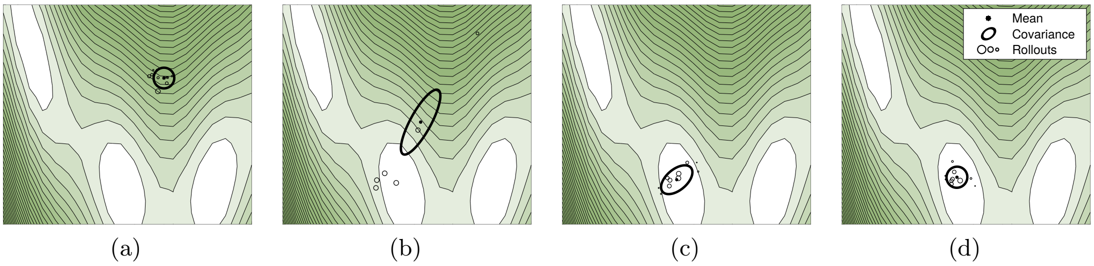
$$\boldsymbol{q}_{t+1} = \boldsymbol{q}_t + \Delta t \cdot \Delta \boldsymbol{q}_t, \tag{2.27}$$

with vector field represented as ELM, given by $\Delta \hat{\boldsymbol{q}}_T = ELM(\boldsymbol{q}_t)$ according to Equation 2.1. Convergence to fixed point attractors is realized by satisfying asymptotic stability constraints defined by Lyapunov and result in a constrained optimization problem. Further information on the solution of the quadratic program and an analysis for point-to-point movements can be found in [Lemme et al., 2014].

Optimization

In this work, optimization of two different modalities is performed. First, the robot has to optimize the executed trajectory in relation to a reward function. The reward function encodes the goal of the current task and no a priori knowledge about the reward function nor a gradient is available to adapt the executed motions with respect to the returned reward. Therefore, optimization of the motions of the robot is treated as a Black-Box-Optimization (BBO) problem. In [Stulp and Sigaud, 2013], an overview and an extended discussion about prominent optimization methods for BBO can be found. Further, a discussion about current methods and trends in reinforcement learning is presented in [Sigaud and Stulp, 2018]. This work does not aim at a comparison of different optimization methods and relies for optimization on the prominent CMA-ES algorithm, as introduced in the following. The second optimization problem aims at controlling the joints of highly compliant robots and compensating for dynamics effects caused by e.g. interaction with the environment. Iterative Learning Control (ILC) can be used to generate a forward signal to support the low-level controller of the joints. ILC is based on the idea, that the performance of a system can be improved by learning from previous executions. It aims at an efficient reduction of the tracking error of a commanded trajectory and the resulting trajectory of the joints. Gradient information of the control signal w.r.t. the torque and the resulting acceleration of the actuator is used to iteratively update a forward signal to minimize the tracking error. A similar method for optimization of forward signals and error minimization was proposed as Repetitive Control (RC) [Hillerström and Walgama, 1996], which is applied to continuous processes with a periodic input and operates in frequency space. Further details of CMA-ES & ILC are presented in the following:

a) Black-Box Optimization (BBO): Covariance Matrix Adaptation - Evolutional Strategy (CMA-ES)



(a) (b) (c) (d)

Figure 2.5: CMA-ES algorithm example on the Branin function. Optimization is visualized for the first, third, sixth and seventh generation.

The original algorithm of CMA-ES relies on four main steps, detailed information can be found in [Hansen, 2006]. Optimization is performed in generations, which means that an action has to be performed under several perturbations. The mean estimate is updated based on the observation of rewards of the executed actions. CMA-ES has an internal representation of the current mean and of the covariance matrix that allows for sampling of new actions normally distributed around the current mean estimate. In addition, CMA-ES estimates an evolution path for the mean and the covariance matrix update. Those evolution paths allow for more stability to outliers and noise. The first step performs the sampling from a multivariate normal distribution centered at the current estimate $\mathbf{m}^{(g)}$, given by

$$\begin{aligned}\mathbf{x}_k^{(g+1)} &= \mathbf{m}^{(g)} + \sigma^{(g)}\mathbf{y}_k^{(g+1)}, \quad \text{with} \\ &\qquad \mathbf{y}_k^{(g+1)} \sim \mathcal{N}_k(\mathbf{0}, \mathbf{C}^{(g)}) \quad \text{for } k = 1, ..., \lambda.\end{aligned} \tag{2.28}$$

Followed by the update of the estimated solution for the next generation, $g+1$, with respect to the rewards of the sampled rollouts

$$\mathbf{m}^{(g+1)} = \mathbf{m}^{(g)} + \sigma^{(g)}\mathbf{y}_{\mathbf{w}}^{(g+1)}, \quad \text{with } \mathbf{y}_{\mathbf{w}}^{(g+1)} = \sum_{i=1}^{\mu} \mathbf{w}_i \mathbf{y}_{i:\lambda}^{(g+1)}. \tag{2.29}$$

The vector $\mathbf{x}_{i:\lambda}^{(g+1)}$ denotes the i-th best individual and the index $i:\lambda$ denotes the index of the i-th ranked individual $R(\mathbf{x}_{1:\lambda}^{g+1}) \leq R(\mathbf{x}_{2:\lambda}^{g+1}) \leq ... \leq R(\mathbf{x}_{\lambda:\lambda}^{g+1})$. With current mean $\mathbf{m}^{(g)}$ and covariance $\mathbf{C}^{(g)} \in \mathbb{R}^{FxF}$ scaled by $\sigma^{(g)} \in \mathbb{R}_+$ for generation g. The third step targets the update of the covariance matrix

$$\begin{aligned}\mathbf{C}^{(g+1)} = (1 - c_1 - c_\mu)\mathbf{c}^{(g)} + c_1 \mathbf{p}_c^{(g+1)} \mathbf{p}_c^{(g+1)^\top} \\ + c_\mu \sum_{i=1}^{\mu} \mathbf{w}_i \mathbf{y}_{i:\lambda}^{(g+1)} \mathbf{y}_{i:\lambda}^{(g+1)^\top}\end{aligned} \tag{2.30}$$

and its evolution path

$$\mathbf{p}_c^{(g+1)} = (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{1 - (1 - c_c)^2}\sqrt{\mu_{eff}}\mathbf{y}_{\mathbf{w}}. \tag{2.31}$$

The final step performs an update of the exploration width sigma,

$$\sigma^{(g+1)} = \sigma^{(g)} \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{||\mathbf{p}_\sigma^{(g+1)}||}{\mathbb{E}||\mathcal{N}(\mathbf{0}, \mathbf{I})||} - 1\right)\right) \tag{2.32}$$

and its assigned evolution path $\mathbf{p}_\sigma^{(g+1)}$,

$$\mathbf{p}_\sigma^{(g+1)} = (1 - c_\sigma)\mathbf{p}_\sigma^{(g)} + \sqrt{1 - (1 - c_\sigma)^2}\sqrt{\mu_{eff}}\mathbf{C}^{(g)^{-1/2}}\mathbf{y}_{\mathbf{w}}^{(g+1)}. \tag{2.33}$$

The operation performed by $\mathbf{C}^{(g)^{-1/2}}$ results in a rescaling of the expected distance of samples to the center, as described in [Hansen, 2006] (Eq. 23).

Figure 2.5 shows a typical behavior of CMA-ES. For illustration, optimization of the *Branin Function* was performed. Starting at a random position in the function space Figure 2.5a, the covariance of the estimated mean first shapes into the direction of the gradient Figure 2.5b and starts shrinking as soon the mean estimate approaches a maximum of the objective function, shown in Figure 2.5c-2.5d.

b) Forward Signal Optimization: Iterative Learning Control (ILC)

Iterative Learning Control (ILC, [Arimoto et al., 1984; Longman, 1998; Norrloff and Gunnarsson, 2002; Wang et al., 2009]) can be applied for optimization

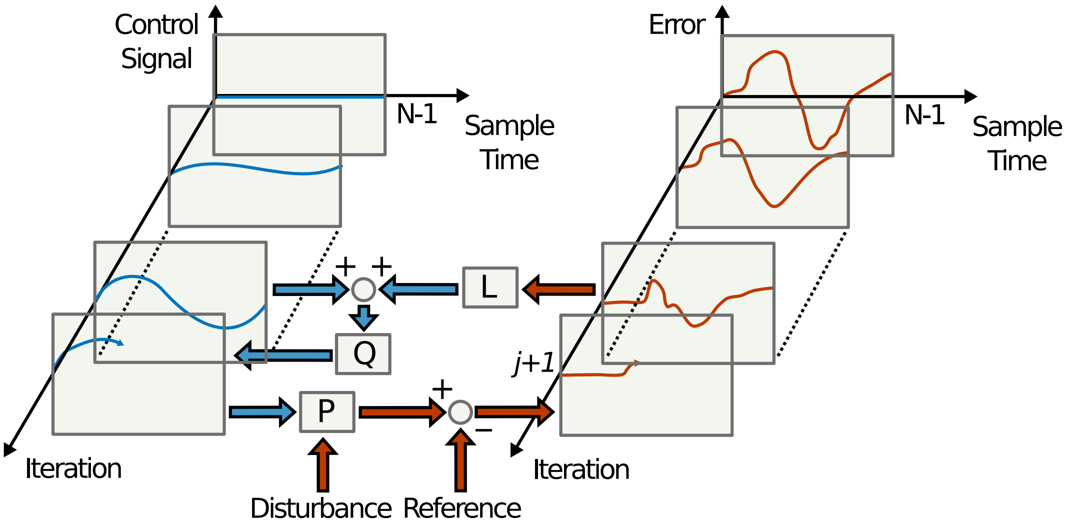


Figure 2.6: Illustration of the Iterative Learning Control (ILC) algorithm. The forward signal is updated according Q-Filter and learning function L, the error signal is estimated by execution of a reference trajectory on plant P. Redrawn from [Bristow et al., 2006].

of feed-forward signals to support the feedback controller and to compensate for repetitive disturbances. A survey of multiple variants of ILC is presented in [Bristow et al., 2006]. Initially proposed as a solely feed-forward approach, ILC was later applied in combination with feedback control as well, like in [Roover and Bosgra, 2000; Bristow et al., 2006]. A successive observation and update of the feed-forward signal leads to a reduction of the tracking error and thereby to a lower feedback controller response. An illustration of the iterative update of the forward signal is shown in Figure 2.6. The figure shows the iterative update of the forward signal based on the forward signal of the previous timestep and the current error of the execution on the plant (P). ILC is widely used in industrial application areas, e.g. for enhancing positioning precision of machines [Chen and Hwang, 2005; Kim and Kim, 1996].

The update of the forward signal is based on a Q-Filter and learning function L. A low-pass filter Q suppresses high frequency learning and contributes to the stability of ILC. An ILC learning algorithm for a delay free system, as shown in [Wang et al., 2009], is given by

$$u_{i+1}(t) = Q(u_i(t)) + \underbrace{L(e_i(t))}_{\text{Update Law}} . \tag{2.34}$$

It includes the previous forward signal $u_i(t)$ filtered through a Q-filter Q and an update law given by the L-Filter response L of the tracking error $e_i(t)$ for iteration i at timestep t. Several schemes to design the Q- and L-filters have been proposed. The Q-filter improves the robustness of ILC by suppressing high frequencies but adds a bias for the reduction of the tracking error. One of the most prominent update laws is the PD-learning rule, which is used for the experiments presented in Chapter 6. For the PD-Type learning, the feed-forward signal is updated based on a proportional (P) and derivative (D) gain of the current error. Implementation details will be discussed in Chapter 6.

2.2.3 Platforms and Datasets

Several robotic platforms, datasets and toy functions are used throughout this thesis for evaluation of the proposed methods. The following list in Table 2.2, will give an overview of the robotic platforms. Additionally, an overview of the datasets and the location of their introduction as well as a list of references to experiments that refer to the datasets is presented in Table 2.1.

Introduction		Experiments	
	Picture	Description	Page
COMAN (Simulation)		High DOF humanoid robot platform. Experiments in this thesis are restricted to point reaching tasks on upper body including arms.	53
BHA		Pneumatically driven trunk like robot. Main flexibility results from 3 sections, each consisting of 3 pneumatic chambers. No precise analytical model available due to continuum kinematics.	96
Affetto		Pneumatically driven humanoid robot child with parallel kinematic. Dynamics model not available and feedback control results in imprecise control.	97
UR5 (Simulation)		6-DOF industrial light-weight robot manipulator. Dynamics simulation based on Gazebo simulation environment.	119

Tab. 2.1: Overview of robotic platforms.

	Introduction		Experiments	
	Name	Page	Description	Page
Toy Data	Barnin	34	CMA-ES Illustration	34
	Sine	46	Reward Regularization	45
			Hybrid Optimization:	
	Circle	74	Overshoot Scenario	76
			Distortion Scenario	77
			Failed Overshoot Scenario	79
	Branch	74	Multiple Minima Scenario	78
	Qubic	88	Transfer Learning	88
BHA	Equilibrium	102	Controller Optimization	103
			Interaction Mode	106
UR5 Simulation	Equilibrium	122	Sensitivity Analysis	123
Affetto	Equilibrium	122	Controller Optimization	108
			Teaching Mode	59
	Drumming	58	Skill Learning	59
			Hybrid Optimization	82
			Transfer Learning	59
	Param. Traj.	130	Skill Learning	135
	Baby Toy	139	Skill Learning	144
COMAN	Reaching	53	Skill Learning	56
			Hybrid Optimization	81
Planar-Arm Simulation	Reaching	48	Reg. of Reward	48
	Via-Point	51	Skill Learning	54
			Hybrid Optimization	80
	Param. Traj.	130	Skill Learning	132

Tab. 2.2: Overview of designed datasets and scenarios for evaluation of the proposed methods.

Chapter 3

Parameterized Skills for Kinematic Representations

Chapter Overview *This chapter presents an evaluation of the incremental skill learning architecture for kinematic representations as proposed in Section 2.2. To improve the generalization capabilities of the skill representation an additional optimization constraint is added to the reward function. The constraint is called* regularization of the reward *and its effect on the skill representation is evaluated on toy data and on an inverse kinematics task of a simulated 10-DOF arm. Further, a bootstrapping process is introduced which supports efficient skill learning. Optimization of solutions for unsolved task instances is accelerated by considering the gradually improving solutions that are proposed by the parameterized skill. Evaluation of the bootstrapping process is performed in simulation and real robot experiments.*

This Chapter is Partially Based on:

- Queißer, J. F., R. F. Reinhart, and J. J. Steil
  2016. Incremental bootstrapping of parameterized motor skills. In *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Pp. 223–229
- Schulz, A., J. F. Queißer, H. Ishihara, and M. Asada
  2018. Transfer learning of complex motor skills on the humanoid robot affetto. In *Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE

3.1 Task-Parameterized Skills

As discussed in Section 1.1, many motor skills have an intrinsic, low-dimensional parameterization, e.g. reaching through a grid to different targets. Repeated policy search for new parameterizations of such a skill is inefficient because the structure

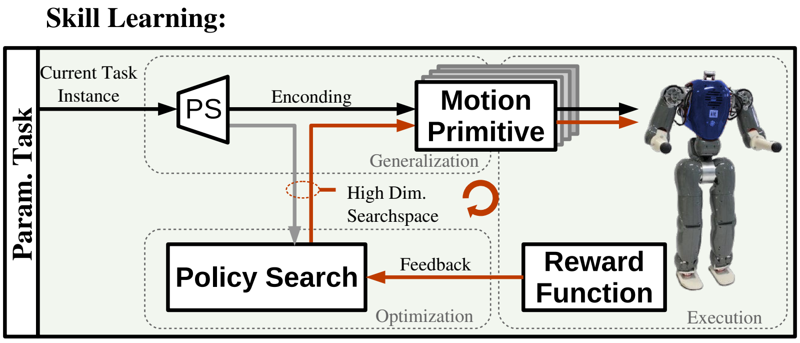


Figure 3.1: Bootstrapping framework, the optimizer is initialized (**H3.2**) by the current estimate (gray) of the parameterized skill and performs an optimization in the high-dimensional space of motion primitives (red). Each executed action (rollout) is assessed by a pre-designed reward function.

of the skill variability is not exploited. This issue has been previously addressed by parameterized skills that provide mappings from task parameters to policy parameters, as discussed in Section 2.1. In this chapter, a bootstrapping technique for the proposed skill learning framework (Section 2.2) is introduced which allows for an efficient and incremental learning of parameterized skills. The approach combines iterative learning with state-of-the-art black-box policy optimization. Further, it is shown that the number of required rollouts can be reduced significantly in case optimization of policies for novel tasks is necessary. Experimental evaluation is based on the success rate of the actions that are generalized by the parameterized skill for unseen task instances and the number of rollouts that are required for the optimization of unsolved task instances.

The work presented in this chapter follows the idea to apply dedicated policy optimization for unseen task parameterizations instead of collecting demonstrations from a tutor as in [Silva et al., 2012; Baranes and Oudeyer, 2013; Silva et al., 2014]. In a similar way as [Baranes and Oudeyer, 2013], a generalization for unseen task parameterizations results in a transfer of optimized results. An incremental algorithm is investigated to establish parameterized skills that reuse previous experience to successively improve the initialization of the optimization process, as previously presented by the author in [Queißer et al., 2016]. Thereby, it is possible to incorporate state-of-the-art optimization of the policy [Sigaud and Stulp, 2018], i.e. by CMA-ES, in comparison to optimization of meta-parameters of policies [Kober et al., 2012] and methods that rely on library-based approaches [Mülling et al., 2010].

In contrast to [Silva et al., 2012, 2014], the optimizer is initialized with the current estimate of the iteratively trained skill which results on an efficient optimization of policy parameters for unsolved task parameters, as outlined in Figure 3.1. Experimental evaluation shows that this leads to a significant reduction of the number of required rollouts during skill acquisition. This thesis refers to the process of incremental skill acquisition as *bootstrapping*. It is systematically shown that the optimization process benefits from the initial condition proposed by the not yet fully trained parameterized skill and how this benefit depends on the model complexity of the learning algorithm. To cope with redundancy and to support the exploration of manifolds with a reduced degree of nonlinearity, an additional cost term is introduced for optimization. This cost term will be referred to as *regularization of the reward*. In addition, ridge regression with regularization is applied for estimation of the parameterized skill. The proposed algorithm for bootstrapping of parameterized skills achieves a significant speed-up of the optimization processes for novel task parameterizations.

The evaluation of the bootstrapping process of the proposed skill learning framework is performed on a via-point task with a planar 10-DOF robot arm (see Figure 3.8). The scalability of the approach is demonstrated by bootstrapping a parameterized skill for a reaching task which is performed on the upper body kinematics of the humanoid robot COMAN (see Figure 3.2) in end effector as well as in joint space control and a drumming task on the humanoid robot platform Affetto. The work introduced in this chapter implements the skill learning for kinematic representations as presented in Section 2.2 and its contribution aims at the experimental verification of the following hypotheses:

H3.1) The generalization capabilities of the parameterized skill benefit from an additional optimization constraint that penalizes solutions that are far-off the current (but faulty) estimate of the parameterized skill. (Section 3.3)

H3.2) Initialization of the optimizer with the current estimate of the parameterized skill leads to a faster optimization and convergence of the skill learning. (Section 3.4)

3.2 Bootstrapping of Parameterized Skills

The presented bootstrapping algorithm results in an efficient skill learning an of a parameterized skill PS($\boldsymbol{\tau}$) by consolidation of optimized policy parameterizations $\boldsymbol{\theta}$ for given task parameterizations $\boldsymbol{\tau}$, according to the formalization in Section 2.2. For this purpose, it is assumed that some sort of policy representation, e.g. a motion primitive model, and policy search algorithm, e.g. REINFORCE [Williams, 1992] or CMA-ES [Hansen, 2006], is available. The idea is to incrementally train the parameterized skill PS($\boldsymbol{\tau}$) with task-policy parameter pairs $(\boldsymbol{\tau}, \boldsymbol{\theta}^*)$, where $\boldsymbol{\theta}^*$ are optimized policy parameters obtained by executing the policy search algorithm for task instance encoded as $\boldsymbol{\tau}$. The key step is that the current estimate PS($\boldsymbol{\tau}$) of policy parameters is used as initial condition for policy optimization of new tasks $\boldsymbol{\tau}$. The most important outcome of this procedure shows that policy search becomes very efficient due to incrementally better initial conditions of the policy search, as stated by hypothesis **H3.2**. Ultimately, PS($\boldsymbol{\tau}$) directly provides optimal policy parameters and no further policy optimization needs to be conducted.

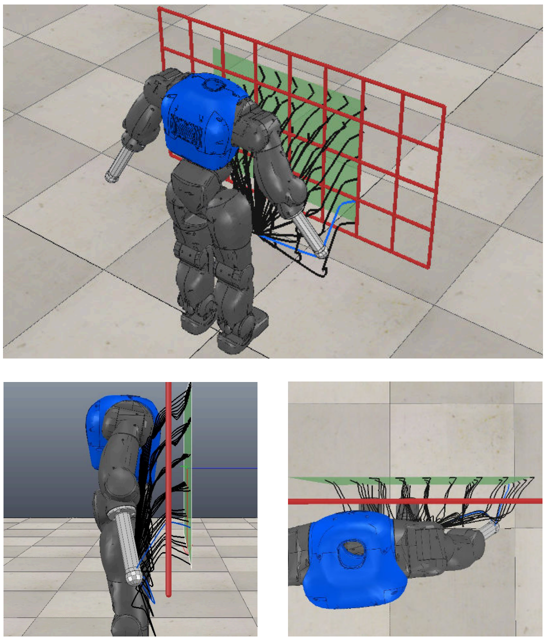
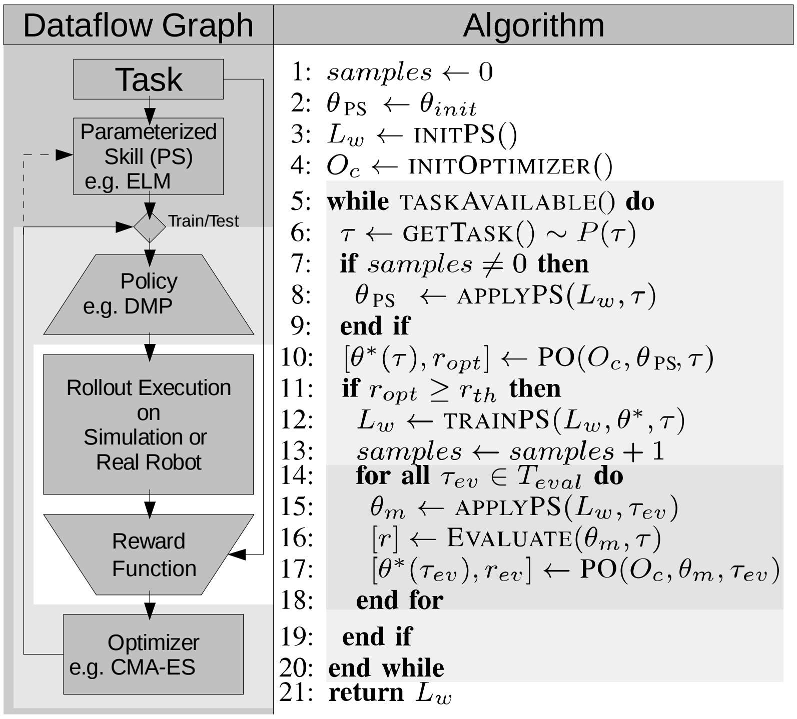


Figure 3.2: Constrained reaching scenario with an upper body of a humanoid robot and a grid-shaped obstacle. Generalized end effector trajectories for different reaching targets that are retrieved from the iteratively trained parameterized skill are shown by black lines.

The algorithm for the parameterized skill acquisition is outlined in Figure 3.3. For each new task $\boldsymbol{\tau}$, the parameterized skill provides an initial policy parameterization $\boldsymbol{\theta}_{\mathrm{PS}} = \mathrm{PS}(\tau)$ (line 8). After collecting a sufficient number of pairs $(\boldsymbol{\tau}, \boldsymbol{\theta}^*)$, the proposed parameterization $\boldsymbol{\theta}_{\mathrm{PS}}$ can achieve satisfactory rewards such that no further Policy Optimization (PO) by reinforcement learning is necessary. In case the estimated policy parameters cannot yet solve the given task or further training is desired, the optimization from initial condition $\mathrm{PS}(\boldsymbol{\tau})$ is initiated (line 10). To ensure that only successful optimization results are used for training of the parameterized skill, an evaluation of the optimization process (e.g. reward r_{opt} exceeds a threshold r_{th}) is performed (line 11). If the optimization was successful, the pair $(\boldsymbol{\tau}, \boldsymbol{\theta}^*)$ with optimized policy parameters $\boldsymbol{\theta}^*$ is used for supervised learning of $\mathrm{PS}(\boldsymbol{\tau})$ (line 12). Finally, lines 14-18 serve evaluation purposes during incremental training. The evaluation was performed on a predefined set of evaluation tasks in $\boldsymbol{\tau}_{ev} \in T_{ev}$ that are disjunct from the training samples.

3.2.1 Component Selection

The following presents a brief introduction of the chosen policy representation and the algorithm for policy optimization and learning that are used throughout this chapter:



```
$samples \leftarrow 0$
$\theta_{\text{PS}} \leftarrow \theta_{init}$
$L_w \leftarrow$ INITPS()
$O_c \leftarrow$ INITOPTIMIZER()
while TASKAVAILABLE() do
  $\tau \leftarrow$ GETTASK() $\sim P(\tau)$
  if $samples \neq 0$ then
    $\theta_{\text{PS}} \leftarrow$ APPLYPS($L_w, \tau$)
  end if
  $[\theta^*(\tau), r_{opt}] \leftarrow$ PO($O_c, \theta_{\text{PS}}, \tau$)
  if $r_{opt} \geq r_{th}$ then
    $L_w \leftarrow$ TRAINPS($L_w, \theta^*, \tau$)
    $samples \leftarrow samples + 1$
    for all $\tau_{ev} \in T_{eval}$ do
      $\theta_m \leftarrow$ APPLYPS($L_w, \tau_{ev}$)
      $[r] \leftarrow$ EVALUATE($\theta_m, \tau$)
      $[\theta^*(\tau_{ev}), r_{ev}] \leftarrow$ PO($O_c, \theta_m, \tau_{ev}$)
    end for
  end if
end while
return $L_w$
```

Figure 3.3: Dataflow and pseudocode of the proposed bootstrapping algorithm. The parameterized skill (PS) estimates a policy parameterization $\boldsymbol{\theta}_{\text{PS}}$. In case of training, successive policy optimization (PO) by reinforcement learning results in an update of the parameterized skill. The shading of the background highlights nested processing loops of the system (from outer to inner): (1) Iteration over all tasks; (2) Optimization of $\boldsymbol{\theta}$ by the PO algorithm; (3) Execution and estimation of the reward by iterating over all T timesteps of the trajectory $\boldsymbol{p}^*_t$.

a) Selection of Policy Representation:

The proposed method does not rely on a specific type of policy representation. Many methods for compact policy presentation have been proposed, e.g. based on Gaussian Mixture Regression (GMR) [Günter et al., 2007] or Neural Imprinted Vector Fields [Lemme et al., 2014], as discussed in Section 2.2.2. This chapter refers to Dynamic Motion Primitives (DMP, [Ijspeert et al., 2013]), because they are widely used in the field of motion generation. DMPs for point-to-point motions are based on a dynamical point attractor system (Equation 2.23) that defines the output trajectory as well as velocity and acceleration profiles. The canonical system is typically defined as $\dot{x} = -\alpha x$ or in this case as a linear decay

$$\dot{x} \leftarrow \begin{cases} -\alpha & \text{if } x \geq \alpha, \\ 0 & \text{otherwise.} \end{cases}, \tag{3.1}$$

as in [Kulvicius et al., 2012]. The shape of the primitive is defined by a disturbance f_{DMP}, as defined in Equation 2.24, where a mixture of K equally distributed Gaussians with fixed variances along the canonical system are used.

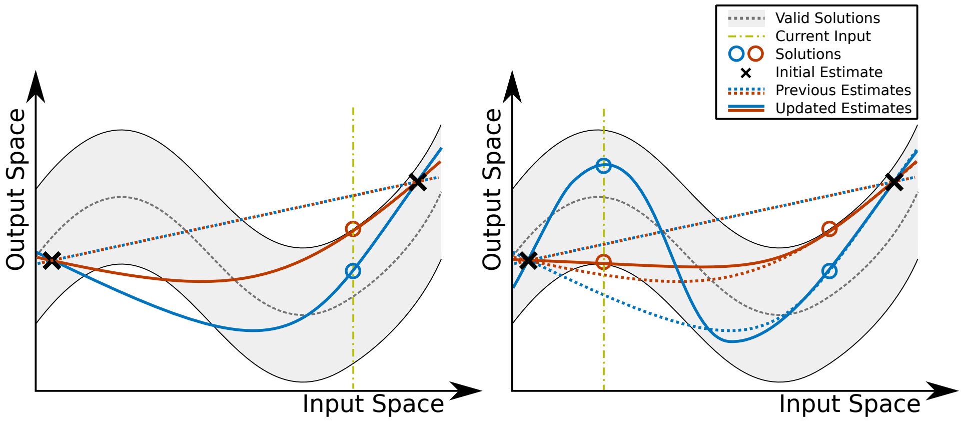
b) Selection of Policy Optimization Algorithm:
For optimization of DMP parameters $\boldsymbol{\theta}^*$ given a task $\boldsymbol{\tau}$, the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES, [Hansen, 2006]) is applied, as introduced in Section 2.2.2. Stulp et al. [Stulp and Sigaud, 2013] have shown that the black-box optimization by CMA-ES is very efficient and reliable in combination with DMPs. In comparison to other reinforcement learning methods like PI^2 [Theodorou et al., 2010] or REINFORCE [Williams, 1992], which evaluate the reward at each time step, CMA-ES operates solely on the total reward of an action sequence. Stochastic optimization by CMA-ES evaluates N_λ rollouts of policy parameters per generation, which are drawn from a Gaussian distribution centered at the current policy parameter estimate. For each generation the current estimate is updated by a weighted mean of all N_λ rollouts. The final number of rollouts R required for optimization is given by the number of generations times the number N_λ of rollouts per generation. Detailed information regarding CMA-ES can be found in Section 2.2.2.

c) Selection of Learning Algorithm:
For learning of parameterized skills $PS(\boldsymbol{\tau})$ an incremental variant of the Extreme Learning Machine (ELM, [Huang et al., 2006]) was implemented. ELMs are feed-forward neural networks with a single hidden layer, thus, the parameterized skill is defined as

$$PS_i(\boldsymbol{\tau}) = \sum_{j=1}^{N_{\text{H}}} \mathbf{W}_{ij}^{out} \sigma(\sum_{k=1}^{E} \mathbf{W}_{jk}^{inp} \boldsymbol{\tau}_k + \boldsymbol{b}_j) \;\; \forall i = 1, ..., F, \tag{3.2}$$

with input dimensionality E, hidden layer size N_{H} and output dimensionality F. Hidden Layer size was set to $N_{\text{H}} = 50$ for generalization in joint space and $N_{\text{H}} = 20$ in case of Cartesian end effector space. Regression is applied on a random projection of the input $\mathbf{W}^{inp} \in \mathbb{R}^{N_{\text{H}} \times E}$, a nonlinear transformation $\sigma(x) = (1 + e^{-x})^{-1}$ and a linear output transformation $\mathbf{W}^{out} \in \mathbb{R}^{F \times N_{\text{H}}}$ that can be updated by incremental least squares algorithms. The incremental update scheme of the ELM was introduced as Online Sequential Extreme Learning Machine (OSELM) [Liang et al., 2006] that incorporates the ability to perform an additional regularization on the weights [Huynh and Won, 2009] or exponential forgetting of previous samples [Zhao et al., 2012]. Since a small number of training data can be expected for skill learning, regularization of the network can help to prevent over-fitting and foster reasonable extrapolation. A more detailed discussion about the learning method and parameter estimation of the readout weights is presented in Section 2.2.2.

3.3 Regularization of Reward



(a) (b)

Figure 3.4: Illustration of the expected effect of the regularization of the reward for the sine-wave experiment. Regularized solutions (red) are expected to result in a smoother memory representation compared to solutions of the non-regularized reward (blue). Two successive learning steps after consolidation of three (a) and four (b) training samples are shown. Range of valid solutions is indicated as gray area. Note, regularization of $\mathbf{W}_{\mathrm{out}}$ is assumed to be equal for both cases.

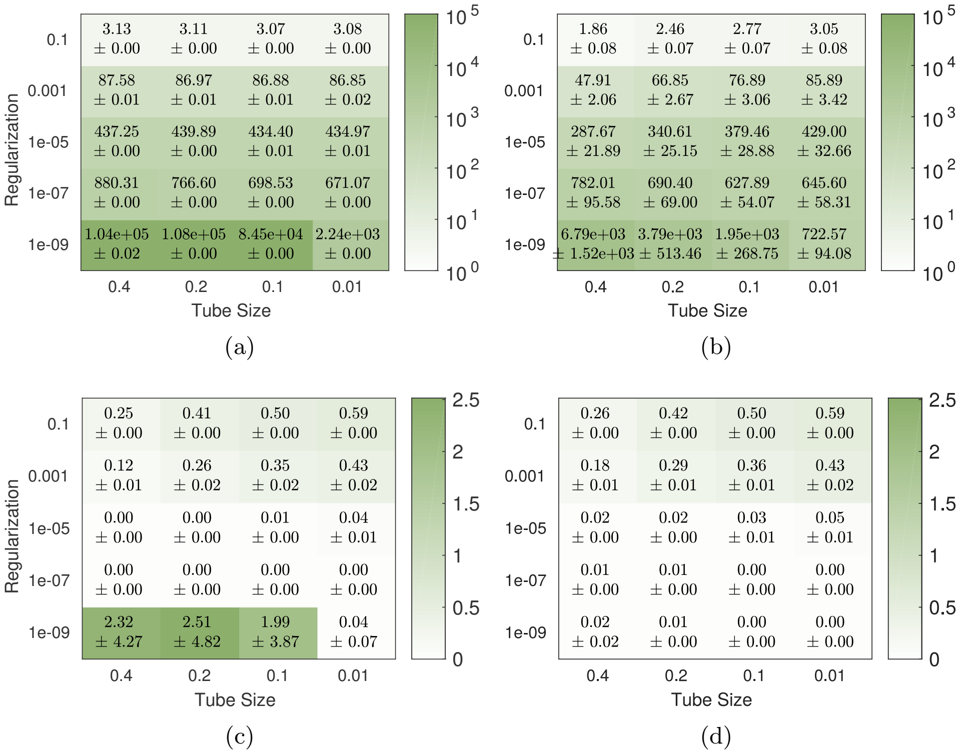
The design of the reward function for successful stochastic optimization of parameterized skills is one of the major challenges. The reward function has a direct influence on the robot's action in relation to the observable variables of the task. In the case of robotic experiments in complex environments, expert knowledge and careful design is a key element for classical reinforcement learning. To avoid explicit modeling of reward functions, alternative approaches propose to learn reward functions automatically or based on expert ratings, like [Daniel et al., 2015] for grasping movements. Inverse reinforcement learning [Ng and Russell, 2000] and minimization of surprise by temporal prediction [Kober and Peters, 2012] are further options to model a target for optimization. The acquisition of parameterized skills relies on the results of the optimized reward function. For the presented framework of this thesis, the parameterized skill is trained with successful solutions gathered by optimization and has to generalize to new task instances. For complex tasks, redundancy in the motor space can be expected as many actions may result in valid task execution. But a high variance of the optimized solutions used for training results in a degraded generalization capability of the parameterized skill.

This section presents an argumentation and a method for a preference of solutions that lie as close as possible to the current estimate of the parameterized skill, as stated in hypothesis **H3.1**. A minimization of the distance of the current estimate $\boldsymbol{\theta}$ to the initialization of the policy search $\boldsymbol{\theta}_{\mathrm{PS}} = PS(\boldsymbol{\tau})$ restricts the variance and the space of successful solutions to be close to the initial estimate. Such solutions result in less adaptation during training and therefore in a lower model complexity of the fully trained parameterized skill. Related work that investigates the effects of regularization in the context of CMA-ES can be found in [Dehio et al., 2016] and shows benefits of an additional objective that minimizes torques for the optimization of mixtures of torque controllers.

By selecting a model with a lower complexity, better generalization capabilities for real world tasks can be expected in the spirit of William of Ockham, known as Ockham's Razor [Jefferys and Berger, 1992].

For the proposed skill learning in comparison to a classical learning problem, the optimizer iteratively selects the training set by a maximization of the reward function. The proposed additional optimization constraint $||\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathrm{PS}}||$ prefers solutions close to the current estimate, the following experiments show that this introduces a heuristic to select incremental training samples in a way, such that the variance of the estimated training set is reduced. In the following, this optimization constraint will be referred to as *regularization of reward*. Note, that the term regularization differs in this context from the common definition of regularization in the context of machine learning as in [Girosi et al., 1995]. But by adding a further term to the reward function, an additional bias is introduced. For the following experiments of the proposed skill learning architecture, the weighting factor for the regularization of the reward is selected in a way that the normalization is approximately one magnitude smaller than the goal of the main objective. By doing so, the optimizer minimizes the distance to the current estimate without a strong disturbance of the original goal.

Experiments Targeting the Model Complexity To evaluate the effects of the regularization of the reward function, an experiment with a simplified toy data set was conducted. The goal for the memory is to learn a parameterized policy represented as a 1D function given by $\mathrm{PS}^*(\tau) = sin(40 \cdot \tau) \pm \omega$. Due to the parameter γ, multiple solutions for a given parameterization of PS^* can be found. The memory was randomly initialized, in case of the first experimental condition, one random configuration τ and its solution were selected from PS^* for each presented training sample. For the second condition that simulates the regularization of the reward function, the memory is trained with solutions of the optimization that are limited to the point with the minimal distance to the current estimate of the memory. Training was performed iteratively and for each training sample, the parameterized skill provided the current estimate based on the previously consolidated training samples. Figure 3.4 illustrates the expected effect of the regularization of the reward function. Figure 3.4a and Figure 3.4b show two successive training states of the memories. The black cross indicates the first two training samples presented to the memories. The current task parameterization is highlighted by dashed vertical red line, the selected training samples are indicated by a colored circle. Depending on the optimization strategy, the non-regularized reward function can end up at any



(a)

Regularization	0.4	0.2	0.1	0.01
0.1	3.13 ± 0.00	3.11 ± 0.00	3.07 ± 0.00	3.08 ± 0.00
0.001	87.58 ± 0.01	86.97 ± 0.01	86.88 ± 0.01	86.85 ± 0.02
1e-05	437.25 ± 0.00	439.89 ± 0.00	434.40 ± 0.01	434.97 ± 0.01
1e-07	880.31 ± 0.00	766.60 ± 0.00	698.53 ± 0.00	671.07 ± 0.00
1e-09	1.04e+05 ± 0.02	1.08e+05 ± 0.00	8.45e+04 ± 0.00	2.24e+03 ± 0.00

(b)

Regularization	0.4	0.2	0.1	0.01
0.1	1.86 ± 0.08	2.46 ± 0.07	2.77 ± 0.07	3.05 ± 0.08
0.001	47.91 ± 2.06	66.85 ± 2.67	76.89 ± 3.06	85.89 ± 3.42
1e-05	287.67 ± 21.89	340.61 ± 25.15	379.46 ± 28.88	429.00 ± 32.66
1e-07	782.01 ± 95.58	690.40 ± 69.00	627.89 ± 54.07	645.60 ± 58.31
1e-09	6.79e+03 ± 1.52e+03	3.79e+03 ± 513.46	1.95e+03 ± 268.75	722.57 ± 94.08

(c)

Regularization	0.4	0.2	0.1	0.01
0.1	0.25 ± 0.00	0.41 ± 0.00	0.50 ± 0.00	0.59 ± 0.00
0.001	0.12 ± 0.01	0.26 ± 0.02	0.35 ± 0.02	0.43 ± 0.02
1e-05	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.04 ± 0.01
1e-07	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
1e-09	2.32 ± 4.27	2.51 ± 4.82	1.99 ± 3.87	0.04 ± 0.07

(d)

Regularization	0.4	0.2	0.1	0.01
0.1	0.26 ± 0.00	0.42 ± 0.00	0.50 ± 0.00	0.59 ± 0.00
0.001	0.18 ± 0.01	0.29 ± 0.01	0.36 ± 0.01	0.43 ± 0.02
1e-05	0.02 ± 0.00	0.02 ± 0.00	0.03 ± 0.01	0.05 ± 0.01
1e-07	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
1e-09	0.02 ± 0.02	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

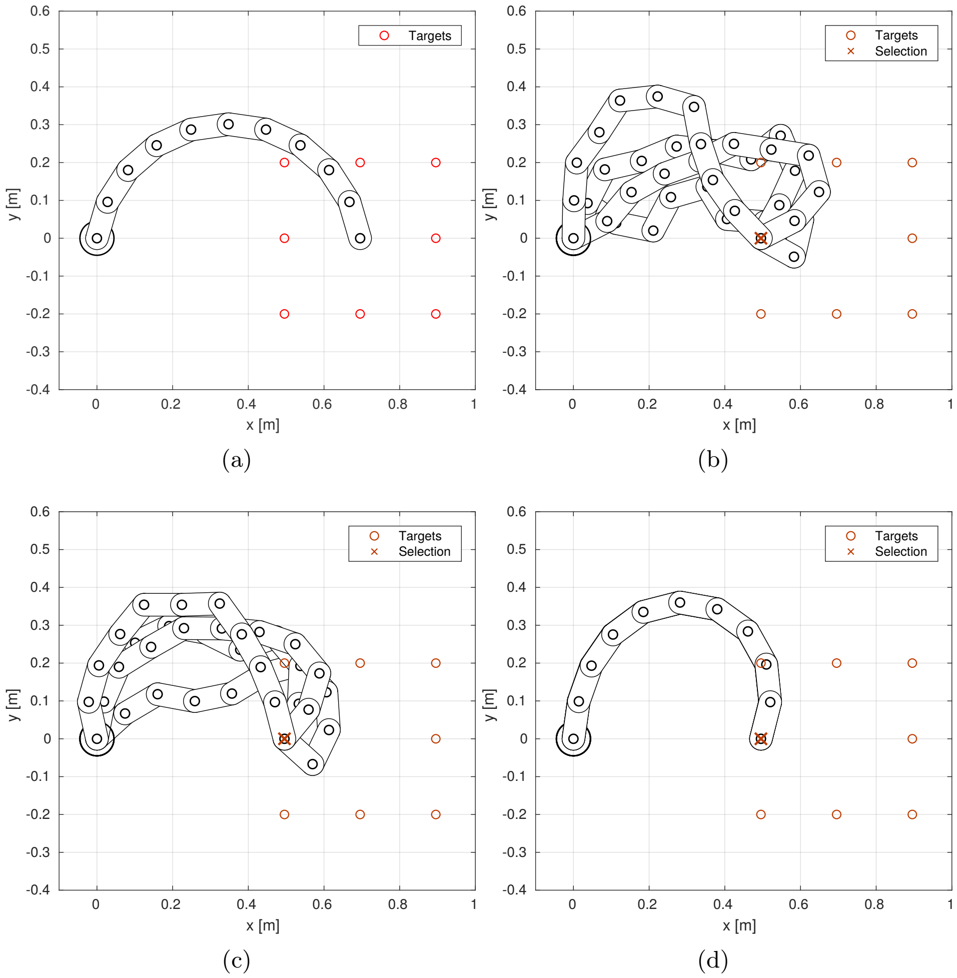
Figure 3.5: Evaluation of the regularization of the reward for the sine-wave experiment. Evaluation is performed for a set of tube sizes and regularization γ of the readout weights by ridge regression. The upper row shows $||\mathbf{W}_{\text{out}}||$, the norms of the readout weights. The second row evaluates the mean absolute error of the estimated function approximation with respect to the tube of valid solutions. The columns refer to the experimental conditions, the first column (a+c) shows results of randomly selected training samples in the tube and the second column (b+d) shows results in case sampling in the tube with the lowest distance to the estimate of the learner is performed.

parameterization of the output function (e.g. blue circle), that fulfills the task. The tube given by ω represents (for this simplified 1D case) various solutions of the optimization in a high-dimensional state space of the robot. In the condition of the regularization of the reward, the optimizer prefers a solution (red circle) that is as close as possible to the previous estimate (gray line), as shown in Figure 3.4a. For evaluation, the regularization of the learner γ as well as the size ω of the range of acceptable solutions was modified. Due to the additional regularization of the readout weights, the degree of nonlinearity of the parameterized skill is reduced. This can be seen for the fourth presented training sample in Figure 3.4b. In the case of the regularization of the reward, the optimizer prefers solutions close to the estimate of the parameterized skill and selects a training set with a lowered degree of nonlinearity. Whereas, the optimizer can estimate a random solution in the case of no regularization of the reward function, which can lead to a higher nonlinearity of the presented training samples to the memory. The results of the experiment can be seen in Figure 3.5. The memory was trained with $N_{train} = 100$ randomly selected samples. For memory implementation, an online sequential ELM with regularization, as introduced in Section 2.2.2, with $N_{\mathrm{H}} = 300$ hidden nodes was utilized. In the first case (Figure 3.5a+c), learning was performed without selecting the closest solution to the current estimate. Whether the second case (Figure 3.5b+d) refers to the closest solution to the current estimate of the learner for training. In case the learners underfit the estimation due to a high regularization ($\gamma = 0.1$) of the output weights W_{out}, both learning methods achieve similar performance. For regularizations of 10^{-5} to 10^{-7} the mean absolute error reaches a level below 10^{-3} and learning was successful for both methods. In case of a low regularization of the readout weights, the training without regularization of the reward suffers from overfitting for wide tube sizes ($\omega \geq 0.1$), as error levels reach 1.99 and more. As illustrated in Figure 3.4, selecting randomly solutions in an interval of valid solutions induces a higher variance of the training data which supports overfitting. In case of the solutions obtained by regularization of the reward, smoother function approximation can be observed due to the selection of solutions close to the parameterized skill. The norms of W_{out} support this observation, in particular in the case of overfitting: the resulting readout weights for the regularized reward function are lower than those for the learner that was trained with random solutions.

Evaluation for 10-DOF Planar Arm The second experiment that investigates the relationship between the regularization of the reward for learning of parameterized skills was performed on a simulation of the kinematics of a 10-DOF planar robot arm. This experiment investigates the effect of the regularization of the reward on the variance of found solutions as well as as the effect on the generalization capabilities the parameterized skill. For simplification, only static postures of the robot are evaluated, i.e. $\pi_{\boldsymbol{\tau}} = \boldsymbol{\tau}$. The task is to estimate a parameterized skill that represents the inverse kinematics of the robot's end effector. Due to the high redundancy of the robot, multiple optimal solutions for one goal positions exist. The home configuration (simulation of $\boldsymbol{\theta}_{\mathrm{PS}}$) of the robot simulation is shown in Figure 3.6a. Optimization for one of the goal positions (marked by red cross) is performed by CMA-ES in joint space. The optimization is initialized by the home configuration with an additional disturbance. The reward is given by:

$$R(\boldsymbol{\theta}, \boldsymbol{v}) = - \underbrace{\|\boldsymbol{p}_{\boldsymbol{\theta}} - \boldsymbol{v}\|^2}_{\text{Target Point (a)}} - \underbrace{\alpha \|\boldsymbol{\theta}_{PS} - \boldsymbol{\theta}\|}_{\substack{\text{Regularization of} \\ \text{Reward (b)}}} , \tag{3.3}$$

with end effector position $\boldsymbol{p}_{\boldsymbol{\theta}}$ for joint configuration $\boldsymbol{\theta}$. The weighting factor $\alpha \in \{0, 0.001, 0.01, 0.1\}$ varies the influence of the regularization of the reward during optimization. As before, the online sequential ELM with regularization, as proposed



(a)

(b)

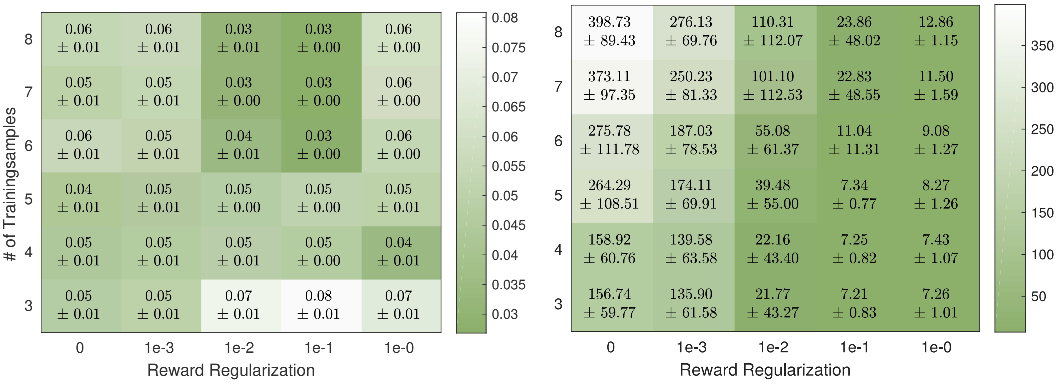
(c)

(d)

Figure 3.6: Regularization of reward, 10-DOF planar arm experiment. Home configuration (a), solutions with a regularization of the reward of zero (b), 10^{-3} (c) and 10^{-2} (d).

in [Huynh and Won, 2011], was utilized with $N = 30$ and a fixed regularization for ridge regression of $\gamma = 10^{-5}$. The Euclidean distance to the initial configuration of the robot (shown in Figure 3.6a) is minimized. Figure 3.6b-c shows ten solutions for one selected goal position with different regularizations of the reward. It can bee seen that the variance of the solutions gets lower, the higher the weighting factor α is selected. In the case of a strong regularization of $\alpha = 10^{-2}$, only one selected solution can be seen due to the high similarity of the overlapping solutions. Additionally, it can be seen that an increasing α leads to a visually more similar appearance to the initial posture due to the representation in joint space, Figure 3.6b-d.

A detailed evaluation can be found in Figure 3.7, it shows the evaluation of the generalization performance of a parameterized skill trained with the solutions of the optimization process. The first part, Figure 3.7a, shows the evaluation of the end effector accuracy for unseen target positions. The evaluation of the norms $||W_{out}||$ of the learner after training are shown in Figure 3.7b. It can be seen that a moderate regularization leads to an improved performance of the generalization capabilities as well as reduced overall norms of the output weights. In case the regularization of the reward function is too strong, the memory suffers from a bias of the optimized solutions and cannot decrease below a mean error rate of 0.6 as in the case for $\alpha = 0.1$ and $N_{tr} = 8$. Table 3.1 shows a summary of the properties of the learned models for $N_{tr} = 8$ training samples in relation to the strength of the regularization of the reward function. The higher the regularization, the lower the variance of the found solutions of the optimization process. At the same time, a bias by the additional optimization constraint, Equation 3.3b, is introduced as it perturbs the main objective of optimization. The generalization performance benefits from a moderate regularization factor, i.e. $\alpha = 10^{-1}$, a compromise between a low bias for optimization and an improved representation of the parameterized skill.



# of Trainingsamples	0	1e-3	1e-2	1e-1	1e-0
8	0.06 ± 0.01	0.06 ± 0.01	0.03 ± 0.01	0.03 ± 0.00	0.06 ± 0.00
7	0.05 ± 0.01	0.05 ± 0.01	0.03 ± 0.00	0.03 ± 0.00	0.06 ± 0.00
6	0.06 ± 0.01	0.05 ± 0.01	0.04 ± 0.01	0.03 ± 0.00	0.06 ± 0.00
5	0.04 ± 0.01	0.05 ± 0.01	0.05 ± 0.00	0.05 ± 0.00	0.05 ± 0.01
4	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.00	0.04 ± 0.01
3	0.05 ± 0.01	0.05 ± 0.01	0.07 ± 0.01	0.08 ± 0.01	0.07 ± 0.01

(a)

	0	1e-3	1e-2	1e-1	1e-0
8	398.73 ± 89.43	276.13 ± 69.76	110.31 ± 112.07	23.86 ± 48.02	12.86 ± 1.15
7	373.11 ± 97.35	250.23 ± 81.33	101.10 ± 112.53	22.83 ± 48.55	11.50 ± 1.59
6	275.78 ± 111.78	187.03 ± 78.53	55.08 ± 61.37	11.04 ± 11.31	9.08 ± 1.27
5	264.29 ± 108.51	174.11 ± 69.91	39.48 ± 55.00	7.34 ± 0.77	8.27 ± 1.26
4	158.92 ± 60.76	139.58 ± 63.58	22.16 ± 43.40	7.25 ± 0.82	7.43 ± 1.07
3	156.74 ± 59.77	135.90 ± 61.58	21.77 ± 43.27	7.21 ± 0.83	7.26 ± 1.01

(b)

Figure 3.7: Evaluation of the regularization of reward on the 10-DOF planar arm scenario. Mean error (a) and norm of readout weights $||\mathbf{W}_{\text{out}}||$ (b) in relation to regularization strength α and the number of presented training samples are shown.

3.4 Experimental Evaluation of Bootstrapping

In the following, an evaluation the applicability of the proposed bootstrapping algorithm can be found. Therefore, two scenarios have been designed to assess the bootstrapping of parameterized skills according to the algorithm from Section 3.2 and hypothesis **H3.2**.

Reg. of	Training Data		Generalization	
Reward	Bias (Error)	Variance	Error	Norm of W_{out}
0	0.0006	1.6569	0.0555	398.73
10^{-3}	0.0004	0.7706	0.0569	276.13
10^{-2}	0.0007	0.2068	0.0325	110.31
10^{-1}	0.0005	0.0246	0.0268	23.86
10^{-0}	0.1600	0.0000	0.0609	12.86

Tab. 3.1: Comparison of the effect of the regularization of the reward. With an increasing regularization of the reward (α), the error on the training data increases and the variance of the training data decreases. The error on the test set for unseen postures reaches a minimum for an intermediate regularization of $\alpha = 10^{-2}$ and the norm of the readout weights of the learner decrease with an increasing α.

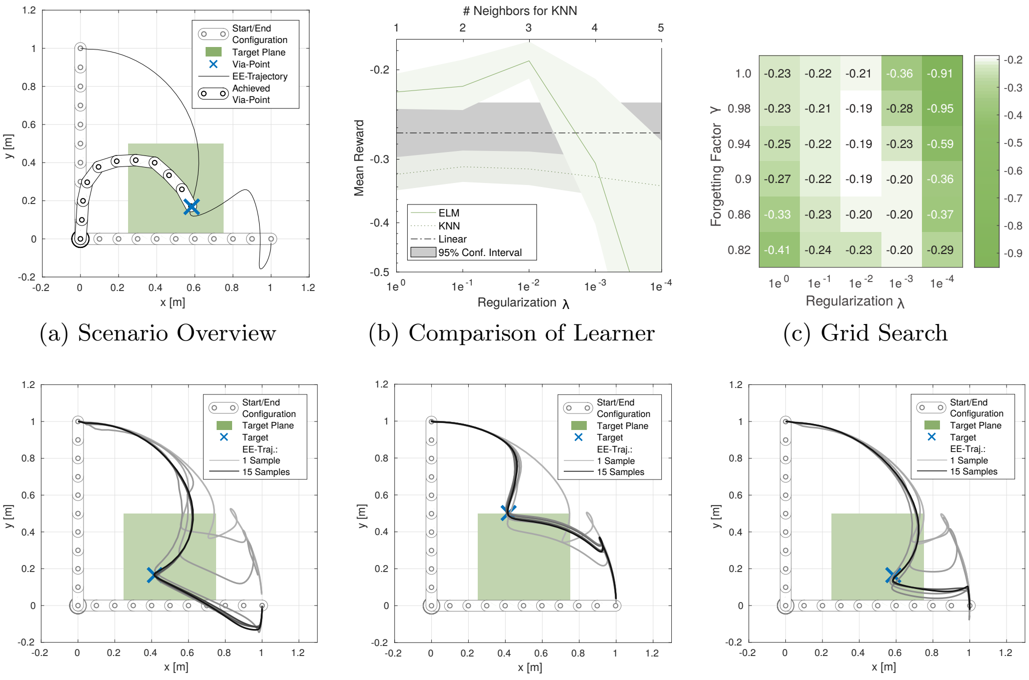
3.4.1 10-DOF Planar Arm Via-Point Task

The goal is to optimize the parameters of a DMP policy to generate joint angle trajectories such that the end effector of the actuator passes through a via-point in task space at time step $\frac{T}{2}$ of the movement with duration T. The experimental evaluation was performed on the kinematics of a 10-DOF planar arm. Motions start at initial configuration $\boldsymbol{q}_{start} = (0,0,0,0,0,0,0,0,0,0)^{\intercal}$ and end at configuration $\boldsymbol{q}_{end} = (\frac{\pi}{2},0,0,0,0,0,0,0,0,0)^{\intercal}$. The task parameterization $\boldsymbol{\tau}$ is given by the 2D via-point position $\boldsymbol{\tau} = (v_x, v_y)^{\top}$ of the end effector at timestep $\frac{T}{2}$.

Since there exists no unique mapping between task and policy parameter space in this example, infinite action parameterizations can be found that sufficiently solve a given task (e.g. exceed a reward threshold). The reward function was extended by a regularization of the reward to reduce ambiguities in the training data for parameterized skill learning. This *regularization of the reward* punishes the deviation of solutions of the optimizer from the initial parameters $\boldsymbol{\theta}_{\text{PS}} = \text{PS}(\tau)$, as discussed in Section 3.3. Further, the reward function prefers a low jerk of the end effector trajectory. The initial and final arm configurations are shown in Figure 3.8a. Initial policy parameters $\boldsymbol{\theta}_{\text{init}}$ have been set to the minimum jerk trajectory [Flash and Hogan, 1984] in joint angle space. The overall reward is given by:

$$R(\boldsymbol{\theta}, \boldsymbol{v}) = \underbrace{-\boldsymbol{\alpha}_1 \sum_{t=2}^{T} \left(\frac{\partial^3 p_{1,t}}{\partial t^3}\right)^2 + \left(\frac{\partial^3 p_{2,t}}{\partial t^3}\right)^2}_{\text{Jerk (a)}} - \underbrace{\boldsymbol{\alpha}_2 \|\boldsymbol{p}_{T/2} - \boldsymbol{v}_p\|^2}_{\text{Via Point (b)}} - \underbrace{\boldsymbol{\alpha}_3 \|\boldsymbol{\theta}_{\text{PS}} - \boldsymbol{\theta}\|}_{\text{Regularization (c)}} \quad (3.4)$$

The reward depends on the DMP parameters $\boldsymbol{\theta}$ that result in a 10 dimensional joint trajectory transformed by the kinematics of the robot arm to the end effector trajectory $\boldsymbol{p}_t$. The jerk is based on the third derivative of the end effector trajectory $\boldsymbol{p}_t$ as proposed in [Fligge et al., 2012] and is represented as one objective of the reward function Equation 3.4a. In addition, the reward function punishes the distance to the desired via-point $\boldsymbol{v}_p = (v_x, v_y)$ of the end effector trajectory (Equation 3.4b) and the regularization term (Equation 3.4c).

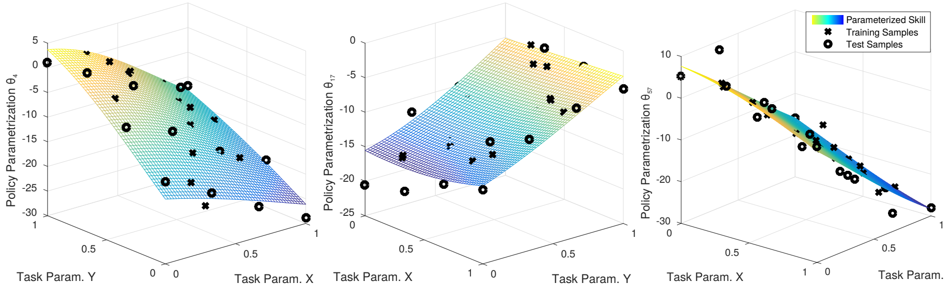


(a) Scenario Overview (b) Comparison of Learner (c) Grid Search

(d) Case I (e) Case II (f) Case III

Figure 3.8: (a) Experimental setup including start/end configuration as well as an optimized solution for one task. (b) Comparison of the generalization of $PS(\boldsymbol{\tau})$ to unseen tasks by linear regression, KNN and ELM with regularization γ. The evaluation shows the mean reward and confidence interval for all test samples $\boldsymbol{\tau}_{ev}$. (c) Forgetting factor evaluation: Mean reward on test samples for $\boldsymbol{\theta}_{\mathrm{PS}}$ after bootstrapping depending on regularization γ and forgetting factor λ. At the bottom (d)-(f), three exemplary test cases for $\boldsymbol{\tau}$ are shown. They show the content of the learned parameterized skill in relation to the number of training samples. The gray scale indicates the number of consolidated training samples.

The coefficients α_i are fixed for all experiments to $\boldsymbol{\alpha} = (10^2, 15, 10^{-3})^\intercal$. The selection of $\boldsymbol{\alpha}$ results in a magnitude of the regularization of ca. 10% of the overall reward of an optimized task, as motivated in Section 3.3. For the training phase $N_{\mathrm{tr}} = 15$ random tasks $\boldsymbol{\tau}$ have been selected, i.e. via-point positions, drawn from the green target plane in Figure 3.8a. Evaluation was done on a fixed test set $\boldsymbol{\tau}_{ev}$ including $N_{\mathrm{te}} = 16$ via-points arranged in a grid on the target plane. For each of the 10 joints of the robot were driven by a DMP with $K = 6$ basis functions, resulting in a $F = 60$ dimensional policy parameterization $\boldsymbol{\theta}$. Figure 3.8d-3.8f shows solutions for three exemplary tasks $\boldsymbol{\tau}$ from the test set. The gray scale of the end effector trajectories refers to the number of consolidated training samples and shows that the



(a) (b) (c)

Figure 3.9: (a)-(c) show three exemplary dimensions of the parameterized skill $PS(\boldsymbol{\tau})$ output in relation to the task parameterization. Task parameterization is the 2D position of the via-point, i.e. $\boldsymbol{\tau} = (v_x, v_y)^\intercal$.

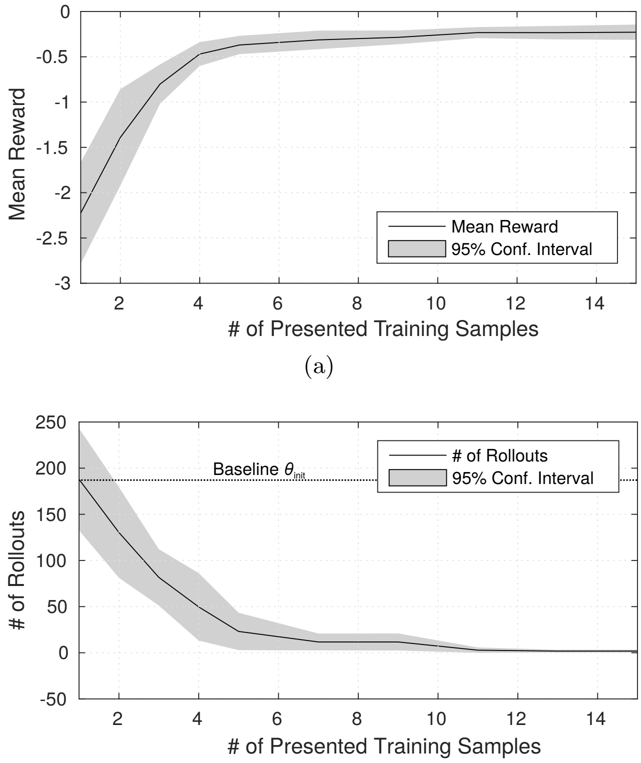
parameterized skill improves as more optimized samples have been used for training. In addition an evaluation of the overall performance that can be achieved by the ELM learner in comparison to KNN Regression and Linear Regression as well as the effect of the regularization of the readout weights was performed. Those results are shown in Figure 3.8b and reveal that the ELM, a nonlinear, global learner for PS($\boldsymbol{\tau}$), is able to gain the highest rewards on the test set.

The effect of an exponential forgetting of training data can be seen in Figure 3.8c. The forgetting factor is implemented by weighted linear regression of the readout weights of the learner of PS($\boldsymbol{\tau}$). By forgetting earlier training samples ($\lambda < 1$), higher rewards can be reached after bootstrapping. As the parameterized skill provides a better initialization for the policy search, better solutions can be found since a better initialization reduces the risk of getting stuck in a local minimum. Therefore it is beneficial to forget earlier solutions in favor of new policy search results. In case not all tasks can be solved by policy search due to local minima (as in Section 3.4.2), an improved initial guess PS($\boldsymbol{\tau}$) can affect the rate of solvable tasks as well.

Figure 3.10a shows the mean initial reward for all tasks $\boldsymbol{\tau}_{ev}$ in the test set for the estimated policy parameters $PS(\boldsymbol{\tau})$ as a function of the number of consolidated training samples. Figure 3.10b shows that policy optimization benefits from the improved initial policy parameters PS($\boldsymbol{\tau}$) by reducing the number of required rollouts to solve novel tasks (exceed a certain reward threshold). A significant reduction of the required number of rollouts compared to the initialization with the first training sample $\boldsymbol{\theta}_{\text{init}}$, i.e. baseline, can be seen.

3.4.2 Reaching Through a Grid

The scenario shows the scalability of the proposed approach to more complex tasks. The goal is to reach for variable positions behind a grid-shaped obstacle while avoiding collisions of the arm with the grid as well as self-collisions. The experiments are performed in simulation of the humanoid robot COMAN [Colasanto et al., 2012] as



(a)

(b)

Figure 3.10: Mean reward of the initial guess $\boldsymbol{\theta}_{\text{PS}} = \text{PS}(\boldsymbol{\tau})$ of the parameterized skill in relation to the number of presented training samples (a) and the mean number of rollouts that are necessary to solve (reward exceeds a threshold) the test tasks (b). Results and confidence interval are based on ten repeated experiments.

shown in Figure 3.2. 7-DOF of the upper body are controlled including waist, chest and right arm joints. For the first part of the experiment, motions are represented in Cartesian space utilizing 3 DMPs with $K = 5$ basis functions (as introduced in Section 3.2.1), resulting in a $F = 15$ dimensional optimization problem. The respective DMPs are executed yielding Cartesian end effector trajectories $\boldsymbol{p}_t^*$. The subset of valid and executable end effector trajectories $\boldsymbol{p}_{r,t}$ in Cartesian space is given by the kinematics as well as the reachability (e.g. joint limits) of the robot joints. For each time step t of the desired end effector trajectory $\boldsymbol{p}_t^*$, an inverse Jacobian controller tries to find a configuration of the robot that complies with $\boldsymbol{p}_t^*$ and maximizes the distance to all obstacles in the null-space of the manipulator Jacobian [Liegeois, 1977]:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger} \left(\boldsymbol{p}_t^* - \boldsymbol{p}_{r,t} \right) + \alpha \left(\mathbb{I} - \boldsymbol{J}^{\dagger} \boldsymbol{J} \right) Z, \text{ with} \tag{3.5}$$

$$Z = \sum_{l=1}^{L} -\boldsymbol{J}_{p,l}^{\intercal} \cdot \boldsymbol{d}_{min,l}. \tag{3.6}$$

The distance $\boldsymbol{p}_t^* - \boldsymbol{p}_{r,t}$ represents the distance between the desired end effector trajectory $\boldsymbol{p}_t^*$ and the trajectory $\boldsymbol{p}_{r,t}$ reached by the robot. The term Z maximizes the distances $||d_{min,l}||$ of all L links to the grid obstacle in the null-space $\mathbb{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J}$. The maximization of the distance to the closest point can be achieved by following the direction $-\boldsymbol{d}_{min,l}$ in joint space by the point Jacobian $\boldsymbol{J}_{p,l}^{\intercal}$ of the closest point to the obstacle. For policy optimization, the reward function is given by

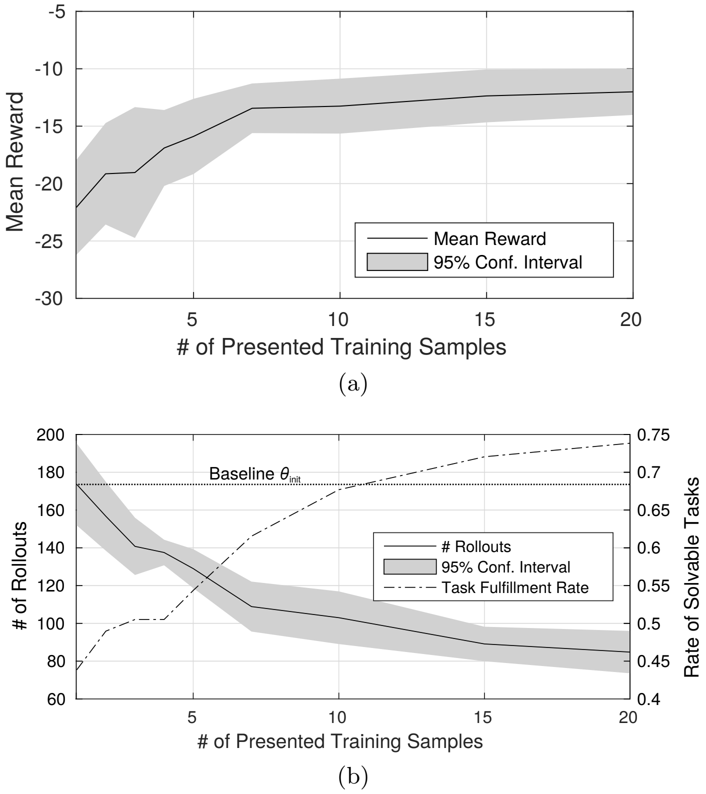
$$R(\theta, \boldsymbol{v}_p) = \underbrace{-\boldsymbol{\alpha}_1 \sum_{t=2}^{T} \|\boldsymbol{p}_t^* - \boldsymbol{p}_{t-1}^*\| -}_{\text{Length of Trajecory (a)}}$$
$$\underbrace{\boldsymbol{\alpha}_2 \sum_{t=1}^{T} \|\boldsymbol{p}_t^* - \boldsymbol{p}_{r,t}\| +}_{\text{Reproducibility (b)}} \underbrace{\boldsymbol{\alpha}_3 \sum_{t=1}^{T} \boldsymbol{r}_{d,t}}_{\text{Dist. to Obstacles (c)}} - \underbrace{\boldsymbol{\alpha}_4 \|\boldsymbol{\theta}_{\text{PS}} - \boldsymbol{\theta}\|}_{\text{Regularization (d)}}, \tag{3.7}$$

with the length T of the trajectory. The reward in Equation 3.7 is a weighted sum of four terms with weighting factors $\boldsymbol{\alpha}_i$: (1) The length of the desired end effector trajectory $\boldsymbol{p}_{d,t}$ that is defined by policy parameter $\boldsymbol{\theta}$; (2) In addition to the punishment of long trajectories (Equation 3.7a), the reward takes the reproducibility of the trajectories into account. Therefore, Equation 3.7b punishes deviations of the reached end effector position $\boldsymbol{p}_{r,t}$ from the desired end effector position $\boldsymbol{p}_t^*$; (3) The distance maximization of all links to the grid obstacle $\boldsymbol{r}_{d,t}$ is considered in Equation 3.7c. The optimization criterion representing the maximization of the distance to the grid-obstacle $\boldsymbol{r}_{d,t}$ is given by

$$\boldsymbol{r}_{d,t} = -\sum_{l=1}^{L} \min\left(0, \|\boldsymbol{d}_{min,l}\| - \boldsymbol{d}_B\right)^2. \tag{3.8}$$

It represents a quadratic relationship to the minimum distances $\boldsymbol{d}_{min,l}$ over all L links to all obstacles in the scene in case the distance falls below a given threshold $\boldsymbol{d}_B$. This criterion refers to the the work presented by Toussaint et al. [Toussaint and Goerick, 2007] where it was used in the context of null-space constraints for humanoid robot movement generation; (4) An additional normalization for small policy parameterizations as given by Equation 3.7d.

The second part of the experiment uses DMPs in joint space to represent the complete motion of the robot. Therefore, the policy parameterization has to represent the maximization of the distance to the grid shaped obstacle implicitly since no additional inverse Jacobian controller is used. This experiment employs seven DMPs with $K = 15$ basis functions (as in Equation 2.24) that generate joint space trajectories, resulting in a $F = 105$ dimensional optimization problem. For policy



(a)

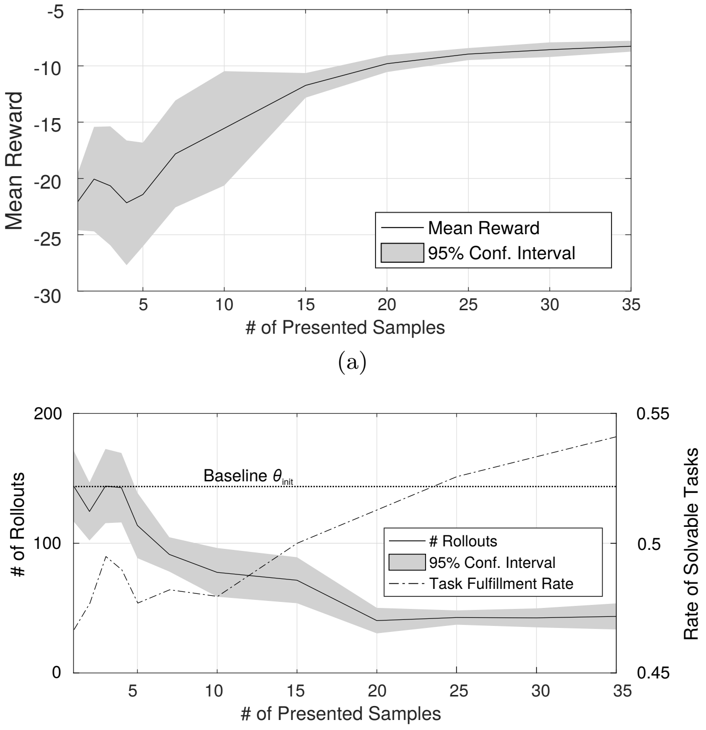
(b)

Figure 3.11: Results of the experiments in Cartesian space. Mean reward of the initial guess $\boldsymbol{\theta}_{\mathrm{PS}} = PS(\boldsymbol{\tau})$ of the parameterized skill in relation to the number of presented training samples (a) and the mean number of rollouts that are necessary to solve selected test tasks (reward exceeds a threshold) (b). The dashed line in (b) shows the mean rate of solvable task in the test set. Results and confidence intervals are based on ten repeated experiments.

optimization, the reward function is similar to the one used for the end effector trajectories Equation 3.7. The policy parameters are decoded by DMPs to desired joint space trajectories $\boldsymbol{p}_t^*$. As previously introduced, Equation 3.7(b) reflects physical constraints of the robot like joint limits. Initial configuration $\boldsymbol{\theta}_{\mathrm{init}}$ is set to joint angle trajectories that allow the end effector to follow a straight line from start to goal position.

Results

An evaluation of the bootstrapping of the parameterized skill was performed, as outlined in Figure 3.3. For training, $N_{train} = 20$ random target positions on the target plane in front of the robot have been selected. For evaluation, a fixed regular grid for point sampling of $N_{test} = 39$ positions on the target plane had been



(a)

(b)

Figure 3.12: Results of the experiments in joint space. Mean reward of the initial guess $\boldsymbol{\theta}_{\mathrm{PS}} = PS(\boldsymbol{\tau})$ of the parameterized skill in relation to the number of presented training samples (a) and the mean number of rollouts that are necessary to solve selected test tasks (reward exceeds a threshold) (b). The dashed line in (b) shows the mean rate of solvable task in the test set. Results and confidence intervals are based on ten repeated experiments.

created. Figure 3.11 reveals that the reward of the initial guess $\boldsymbol{\theta}_{\mathrm{PS}} = PS(\boldsymbol{\tau})$ of the parameterized skill increases with the number of presented training samples. In comparison to the previous experiment in Section 3.4.1, the optimization algorithm does not always succeed to find a solution for all tasks of the test set. Figure 3.11(b) shows an increasing success rate in relation to the number of consolidated samples and thereby the reward of the initial parameters $\boldsymbol{\theta}_{\mathrm{PS}}$ of the policy search. This indicates that increasingly better initial conditions $PS(\boldsymbol{\tau})$ for policy optimization reduce the risk to get stuck in local minima during optimization. In terms of number of rollouts that are required to fulfill a new task, similar results as in the 10-DOF arm experiment can be observed: the number of required rollouts necessary for task fulfillment decreases the more successfully solved task instances have been presented to the parameterized skill as training data. This results in a bootstrapping and acceleration of the parameterized skill learning, as stated by **H3.2**. Although the experiments in cartesian space utilize a joint controller that maximizes the distances automatically, similar performance can be reached in joint space, except of a slightly lower success rate.

3.4.3 Affetto Drumming Scenario

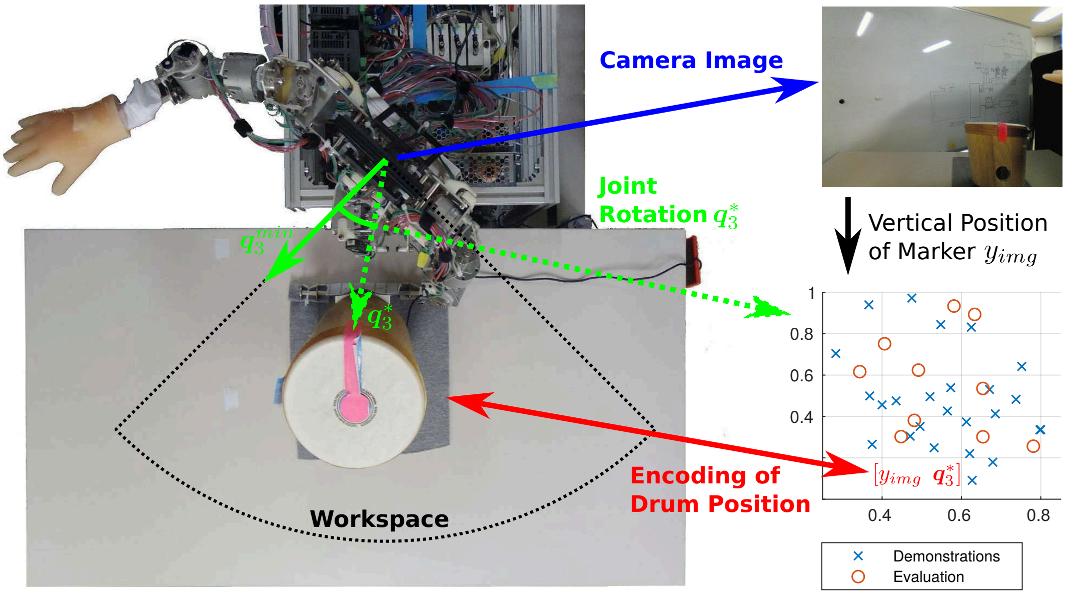


Figure 3.13: Top-down view of the experimental setup of the drumming scenario. Extraction of the low-dimensional task parameterization and the relation to drum position can be seen. Bottom right: training and test set distribution of task parameterization $\boldsymbol{\tau}$.

The following experiment aims at the evaluation of the bootstrapping process for complex robot skills on a real robot system. The upper body of the humanoid robot Affetto has to play a drum placed on a table in front of the robot, as shown in Figure 3.13. For training, the robot is able to observe the drum position directly which results in the task parameterization. Training samples for the parameterized skill are gathered by kinesthetic teaching. Starting from a fixed home position, a human demonstrator moves the arm of the robot in such a way that the hand of the robot hits the drum and a drumming sound is generated. Evaluation of the performance of the parameterized skill is performed by the estimation of the success rate for generalized drumming actions at previously unseen positions of the drum.

The camera attached to the upper body of the robot performs a simple visual search and blob detection of the marker attached to the drum, giving the horizontal $x_{\text{img}} \in [0, 1]$ and vertical $y_{\text{img}} \in [0, 1]$ position of the center, normalized for drum positions in the workspace. To estimate the task parameterization, the robot moves to a fixed starting configuration $\boldsymbol{q}^{\text{start}}$ (shown in Figure 3.14) and centers the marker of the drum in the image of the camera by only rotating the upper body orientation by joint q_3. The joint configuration (see Figure 5.3) of the robot and hardware details are discussed in Section 5.2. The task parameterization $\boldsymbol{\tau} = (y_{\text{img}}, q_3^*)^\top$ includes the final rotation of the upper body q_3^* as well as the height of the marker in the visual image of the camera, resulting in a 2D coordinate that represents the position of the drum relative to the robot. The estimation of the task parameterization is illustrated in Figure 3.13.

Robot Platform The experiments are carried out on the humanoid robot platform Affetto, a pneumatically-actuated highly compliant robot with a 22-DOF upper body structure. The experiments were performed on 8-DOF, including 3-DOF of the abdomen and the right arm and an unactuated soft rubber hand. Policies define joint angle trajectories that are forwarded to the low-level joint controller. To enhance the quality of the tracking performance, the implementation refers to the PIDF controller [Todorov et al., 2010] for the pneumatically driven joints of the robot and optimize the controller parameter by automatic optimization and hand tuning on a test trajectory that includes sine waves and step responses. According to [Todorov et al., 2010], the valve opening is controlled by

$$v_j^+ = k_F(u_j^{\text{PID}} - p_j^{\text{PD}}) \tag{3.9}$$

and vise versa $v_j^- = -v_j^+$ for the antagonistic chamber. Further information regarding the robot platform, the low-level control and parameter estimation can be found in Section 5.3.2.

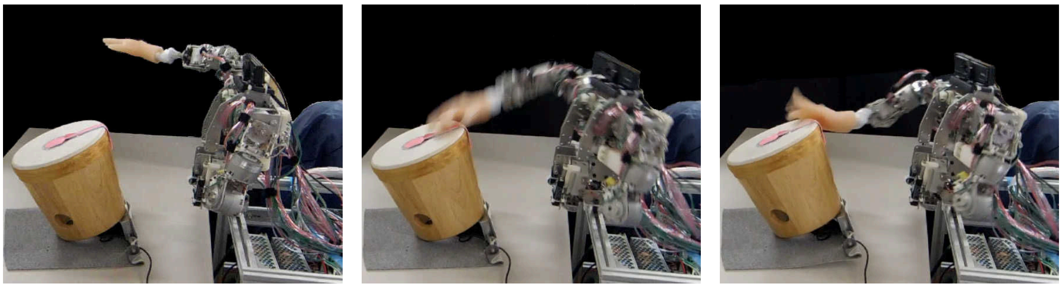
Kinesthetic Teaching Mode To initiate the teaching mode, the joint PIDF controller are commanded to move the joints of the robot to a predefined initial posture $\boldsymbol{q}^{start}$. After convergence of the robot to the initial posture, the control signals u_j^{PID} of the equilibrium states of the joints j are collected as u_j^{eq} and used as an offset for the feedback controller, defined as

$$v_j^+ = k_F(u_j^{\text{PID}} + u_j^{eq} - p_j^{\text{PD}}). \tag{3.10}$$

An equilibrium state of the robot is defined as the state of the robot in which velocity and acceleration are zero, see Section 5.3 for further details. Additionally, the integration of errors is deactivated by setting the integral component I of the controller to zero. It can be expected that u_j^{eq} reflects the integral part of the controller as the proportional and derivative components are zero in equilibrium states. A deflection of the robot joint configuration $\boldsymbol{q}^{start}$ during the demonstration phase results in a counter force given by the feedback controller's proportional gains that aim to move the robot back to its initial configuration. Each trajectory recording is run for 3 seconds and the resulting trajectory is encoded into $\boldsymbol{\theta}$ by the DMPs.

Learning to Drum

The parameterized skill was trained with a collection of successful human demonstrations for $N_{\text{tr}} = 25$ drum positions randomly distributed in the workspace of the



(a) (b) (c)

Figure 3.14: Snapshots of generalized drumming action. Starting configuration $\boldsymbol{q}^{\text{start}}$ is shown in the leftmost picture (a).

robot. Exemplary snapshots for different drum positions from the ego perspective of the robot are shown in Section A.4. In comparison to the previously presented experimental evaluations, no further policy optimization is performed. A demonstration can be considered successful in case the execution of the recorded trajectory by the robot results in a drumming sound. Kinesthetic teaching results in the training set $\mathcal{D} = \{(\boldsymbol{\tau}^k, \boldsymbol{\theta}^k) | k = 1, \ldots, N_{\text{tr}}\}$, which is presented in a random order for an incremental update of the parameterized skill, according to the algorithm presented in Section 3.2. All demonstrations are encoded as a $K = 15$ dimensional DMP for each of the $N_{\text{DOF}} = 8$-DOF of the robot, resulting in a $F = 120$ dimensional parameterization of $\boldsymbol{\theta}$. The reward function is defined based on a distance measure of the recorded audio spectrum to the prototypes, which have been gathered by the execution of training demonstrations. This allows an objective evaluation of the success rate of generalization to unseen drum positions. The similarity measure of a recorded spectrum to one prototype is given by the operator $\circledast : \mathbb{R}^{m \times t_s} \times \mathbb{R}^{m \times t_p} \to \mathbb{R}, \quad (S, P) \mapsto d = S \circledast P$ for input spectrum S, prototype P, m extracted frequency bands and time-steps $t_p \geq t_t$, defined as

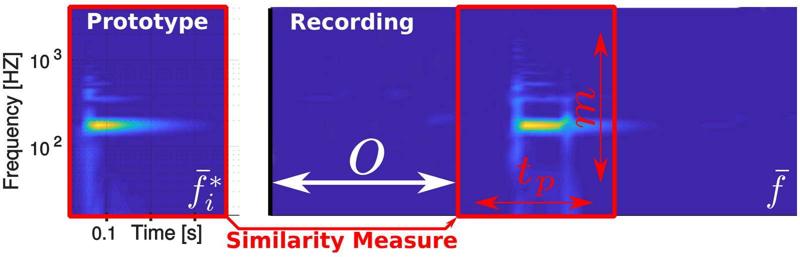


Figure 3.15: Visualization of the similarity measure of spectrograms $\bar{f} \circledast \bar{f}_i^*$.

$$\mathbf{S} \circledast \mathbf{P} \overset{\text{def}}{=} \min_{0 \leq o \leq t_s - t_p} \left(\sum_{i=1}^{m} \sum_{j=1}^{t_p} \left(s(i, j+o) - p(i,j) \right)^2 \right)^{1/2}, \tag{3.11}$$

as visualized in Figure 3.15. The reward function for a recorded spectrum $\bar{f}(\omega, t)$ is given by

$$R(\bar{f}) = \max_{1 \leq i \leq N_{tr}} \frac{\|\bar{f}_i^*\| - \bar{f} \circledast \bar{f}_i^*}{\|\bar{f}_i^*\|}, \tag{3.12}$$

with $\|\bar{f}_i^*\|$ acting as normalization of different prototype activation strengths to a maximum reachable reward of one.

Hidden layer size of the ELM was set to $N_{\mathrm{H}} = 50$ with a regularization $\gamma = 10^{-4}$ for online learning, see Section 2.2.2 for details. Generalization performance was estimated in terms of success rate on a fixed set of $N_{\mathrm{te}} = 10$ positions of the drum that are not part of the training set, as shown in Figure 3.13.

The success rate is estimated by a simple threshold operation on the reward function and counted as successful if $R(\bar{f}) > 0.15$, defined by hand tuning. Figure 3.16 shows the results of the evaluation, it can be seen that the Affetto robot acquires the skill of drumming for all evaluation positions after presentation of all 25 human demonstrations.

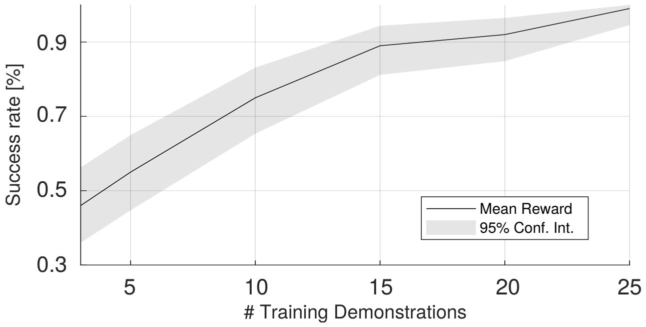


Figure 3.16: Results of the Affetto drumming experiment. Success rate in relation to the number of presented training samples for unseen task instances. Confidence estimate is based on Clopper-Pearson interval.

3.5 Discussion

This chapter introduced a bootstrapping algorithm to incrementally train parameterized skills. Since the optimization of actions has to be performed in real-world scenarios, generation of unseen skills from a small number of training samples is necessary. For that reason, smoothness of the mappings between task and policy parameter spaces can be assumed. The results indicate that the DMP space is well suited for parameterized robot trajectory generation and a smooth mapping between task parameterization and DMP space is a valid assumption. The experimental results verified that the incremental learning of parameterized skills is possible and that the incremental update can significantly speed up policy search for novel task parameterizations, as stated by hypothesis **H3.2**. Moreover, it was shown that initialization of the optimization process with successively improved solutions (i.e. with higher rewards) extends also the number of successfully solved tasks (i.e. exceed a reward threshold).

Additional cost terms for the optimization have been proposed to support consistent training samples without ambiguities caused by the redundancy of the task solutions, which were introduced as regularization of the reward. The experimental evaluation supports the claims of hypothesis **H3.1** as the regularization of the reward resulted in an improved generalization capability, a lower degree of nonlinearity, and decreased output weights of the parameterized skill.

Chapter 4

Efficient Exploration of Parameterized Skills

Chapter Overview *The first part of this chapter presents a novel hybrid optimization method that combines a fast coarse optimization on a manifold of policy parameters with a fine grained parameter search in the unrestricted space of actions. The proposed algorithm reduces the number of required rollouts for adaptation to new task conditions. The application in illustrative toy scenarios, for a 10-DOF planar arm and a humanoid robot point reaching task, validate the approach.*

The second part of this chapter presents a method to reuse knowledge obtained in one situation, in a new related one. This process is known in literature as transfer learning. In order to address such domain adaptation problems, a novel transfer learning algorithm is proposed that maps data from the new domain in such a way that the original model is applicable again. The method is demonstrated on an artificial data set as well as in the robot setting. As a case study, a drumming scenario with the humanoid robot child Affetto is presented in which the environment changes such that the scenario can only be observed through a mirror.

This Chapter is Partially Based on:

- Queißer, J. F., R. F. Reinhart, and J. J. Steil
  2016. Incremental bootstrapping of parameterized motor skills. In *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Pp. 223–229

- Queißer, J. F. and J. J. Steil
  2018. Bootstrapping of parameterized skills through hybrid optimization in task and policy spaces. *Frontiers in Robotics and AI*, 5(49)

- Schulz, A., J. F. Queißer, H. Ishihara, and M. Asada
  2018. Transfer learning of complex motor skills on the humanoid robot affetto.

In *Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE

4.1 Optimization in Task Related Manifolds

As discussed in Chapter 1, modern robotic applications make high demands on adaptation of actions with respect to variance in a given task. Although autonomous robots can perform particularly well at highly specific tasks, learning each task in isolation is a very costly process, not only in terms of time but also in terms of hardware wearout and energy usage. Hence, robotic systems need to be able to adapt quickly to new situations in order to be useful in everyday tasks. In this chapter, a skill learning architecture as proposed in Chapter 3 is assumed: generalization of adapted actions for changing task parameterizations is performed by a parameterized skill, which is encoded as a meta-learner that provides parameters for task-specific dynamic motion primitives. In comparison to the method of the previous chapter that deals with the initialization of the optimization process by the estimate of the parameterized skill, a more efficient optimization process for unsolved task instances is investigated.

The contribution of this chapter is twofold. The first contribution that is presented aims at the introduction of a hybrid optimization method that combines a fast coarse optimization on a manifold of policy parameters with a fine grained parameter search in the unrestricted space of actions. The second contribution investigates the reuse of previous knowledge in case of a changed sensory perception. One way to address this issue is transfer learning, which aims at reusing knowledge obtained in one situation, in a new related one. Evaluation is performed on a drumming scenario for the child robot Affetto. After an initial learning of the skill, the robot is not able to directly observe the drum as before and the robot has to deal with the reflection of the drum in a mirror. In order to address such domain adaptation problems, this chapter introduces a novel transfer learning algorithm that aims at mapping data from the new domain in such a way that the original model is applicable again. The evaluation metric for the proposed algorithm is the number of required rollouts for adaptation to new task conditions. A demonstration of skill transfer is performed on an artificial data set as well as in the robot setting.

The work introduced in this chapter extends the previous method [Queißer et al., 2016] as presented in Chapter 3 and its contribution aims at the experimental verification of the following hypotheses:

H4.1) Optimization in the manifold of previous solutions leads to a reduction of the search space and thereby to a more efficient acquisition of the parameterized skill. (Section 4.2)

H4.2) Skill transfer of an already learned skill allows a more efficient adaptation to changing task condition in comparison of relearning the skill from scratch. (Section 4.3)

4.2 Hybrid Optimization

Based on previous experiments, the presented work deliberates the utilization of the parameterized skill as a mapping from the low-dimensional manifold of task-space to the high-dimensional search space of policy parameters and vice versa, as stated by hypothesis **H4.1**. It is assumed that by performing a policy optimization on this low-dimensional manifold, a further speed-up, in terms of number of rollouts, during the optimization process can be observed. But to cope with the very likely case that no sufficient solution for the required task can be found in the manifold of the parameterized skill, the proposed algorithm performs a hybrid search in both spaces.

Therefore, a novel hybrid optimization algorithm is proposed that samples rollouts in both spaces and performs an estimation of a combined parameter update, as outlined in Figure 4.1.

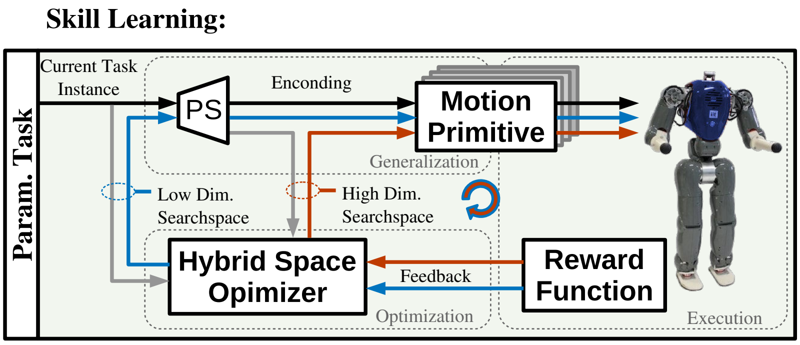


Figure 4.1: Hybrid optimization framework. The optimizer is initialized (**H3.2**) by the current estimate (gray) of the parameterized skill PS and performs a hybrid optimization (**H4.1**) in the low-dimensional manifold of previous solutions (blue) and the high-dimensional space of motion primitives (red).

The evaluation of the hybrid search of the proposed algorithm is performed on the previously introduced via-point task on a planar 10-DOF robot arm (see Figure 3.9).

The scalability of the approach is demonstrated by optimization of a parameterized skill for a reaching task that incorporates the upper body kinematics of the humanoid robot COMAN (see Figure 3.2) in end effector as well as joint space control. The drumming scenario introduced in Section 3.4.3 demonstrates aplicability of the hybrid optimization for complex real robotic scenarios. Additionally, the properties of the proposed optimization in hybrid spaces are elaborated on toy examples.

Related Work Previous work of Koutnik et al. [2010]; Fabisch et al. [2013] has already demonstrated that a compression of the parameter space by use of multi layer perceptrons (MLPs) leads to an acceleration of optimization for reinforcement tasks. Reduction of the search spaces by manifolds for value function approximation [Glaubius and D.Smart, 2005] and abstraction of the whole state-space into sub areas for terrain navigation [Glaubius et al., 2005] can be beneficial in case of reinforcement learning. Constrained optimization problems have been tackled by the reduction of state-space evaluations and a focus on the feasible space of parameters [Ullah et al., 2008]. It was demonstrated that the reduction of the number of available bio-mechanical DOF helps to stabilize the interplay between environmental and neural dynamics for robotic tasks [Lungarella and Berthouze, 2002]. Dimensionality reduction by freezing or synchronization of joints allows for faster skill acquisition, as shown by Kawai et al. [2012]. Further related work has elaborated the intrinsic dimensionality of human movements and demonstrated that dimension reduction is beneficial for reinforcement learning on humanoid robot platforms [Colome et al., 2014].

Hybrid Optimization It is assumed that previously optimized solutions $(\boldsymbol{\tau}, \boldsymbol{\theta}^*)$ represent the variability in the task domain and are consolidated in the parameterized skill. Therefore, the proposed method reconsiders the parameterized skill as an embedding $f_{\mathbf{emb}}$ of a nonlinear manifold of task relevant actions within the full policy space $f_{\text{PS}} \colon \mathbb{R}^E \to \mathbb{R}^F$, $\boldsymbol{\theta}_{\text{emb}} \mapsto f_{\text{PS}} = \text{PS}(\boldsymbol{\theta}_{\text{emb}})$. Further, it is expected that solutions for unseen tasks are located close to the manifold of the parameterized skill, since the relation between a higher number of consolidated samples and a higher initial reward can be observed, as shown in Figure 3.9 and Figure 3.11. Due to the lower dimensionality of the task parameterization compared to the policy parameterization, policy optimization is performed in the input space of f_{PS}.

It can be expected that for points on f_{PS} and their local neighborhood a invertible map, i.e. a chart of the manifold in the policy space, exists. But on a global scale, it can be expected that the mapping between the task space and the policy space is not invertible. Different task parameterizations $\boldsymbol{\tau}$ may require the same policy parameterization $\boldsymbol{\theta}$ and the mapping could not be differentiable due to e.g. joint limits. Previous work related to the proposed method for dimensionality reduction for policy optimization includes primitive based motion generation by PCA compression [Park and Jo, 2004], lower dimensional primitives that encode differences between trajectories [Stulp et al., 2009] and further library based approaches like [Moro et al., 2012].

But clearly, a search in the task space depends heavily on the number and quality of previously seen samples. Finding sufficient solutions for all unseen tasks configurations on a low-dimensional manifold cannot be expected. More specifically, an exploration on the approximated manifold allows for a coarse search that quickly moves the estimation for $\boldsymbol{\theta}^*$ into the direction of higher rewards. If the optimizer is not able to fulfill the given task or is less efficient to find a better solution in the task space, the system is forced to switch to a slower refinement search in the policy space. But also a temporary switch back from a search in the policy space to the task space cannot be excluded.

As optimization in policy space is not bound to the manifold of f_{PS}, the joint update between of both spaces requires an inverse estimate of the parameterized skill. The local inverse of f_{PS} is defined as

$$\widehat{f}_{\mathrm{PS}}^{-1} = \widehat{\mathrm{PS}}^{-1}(\boldsymbol{\theta}) = \min_{\boldsymbol{\tau}} \left\| \mathrm{PS}(\boldsymbol{\tau}) - \boldsymbol{\theta} \right\| , \tag{4.1}$$

which allows to estimate a point on f_{PS} that gives the closest response for a desired output $\boldsymbol{\theta}$ for samples in a local neighborhood of $\mathrm{PS}(\boldsymbol{\tau})$.

The approach allows the combination of rollouts performed in both spaces for an update of the optimization algorithm. The estimation of the importance of each space during optimization is based on the success rate of the policies sampled in their respective spaces, as defined in Section 4.2.2. In general, the combination of optimizers is not limited to a specific optimization algorithm, this work refers to a hybrid CMA-ES approach as introduced in Section 4.2.2.

From a policy optimization that considers both spaces, the following advantages can be expected: First, the algorithm is expected to perform a fast optimization on a low-dimensional manifold followed by an optional successive fine tuning. Second, by exploration of the manifold of the parameterized skill, it can be assumed that solutions that fit to the current estimate of f_{PS} will be found. Therefore, an enhancement of the consistency of the training data of the parameterized skill is expected, in particular for complex reward functions that allow for multiple solutions in policy space. Section 4.2.4-4.2.5 will validate these assumptions. The proposed method will be visualized and discussed on toy data sets and will be compared to to CMA-ES in the full space of the policy parameterization.

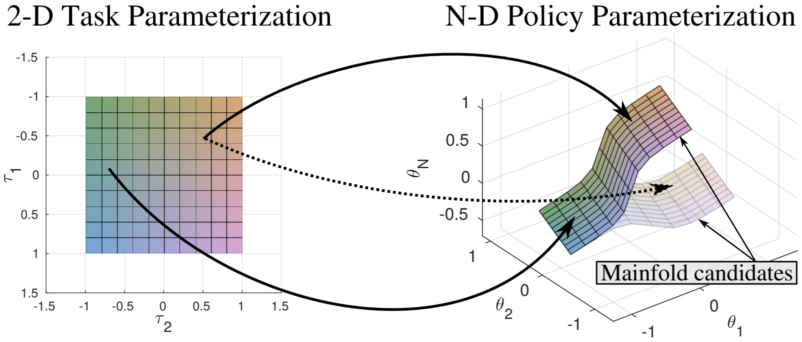


Figure 4.2: It is expected that multiple manifolds exist that are suitable to describe a given task. Therefore the estimation of policy parameterizations that lie close to only one of the manifold candidates allows to estimate a smooth mapping between task and policy parameterization. Policy parameterizations that originate from different manifold candidates can result in ambiguous training data and decrease generalization capabilities of the parameterized skill. Coloring indicates mapping from input space to position on manifold.

Figure 4.2 shows the visualization of the relation between task space and the policy parameterization. For this work, it is assumed that multiple manifolds for a given task parameterization exist. Therefore, one of the candidated manifold for approximation by the parameterized skill has to be selected. The incremental exploration of a continuous mapping between task parameterization $\boldsymbol{\tau}$ and policy parameterization $\boldsymbol{\theta}$ is supported by imposing a respective preference for solutions that are close to the current estimate of the parameterized skill, as discussed in Section 3.3.

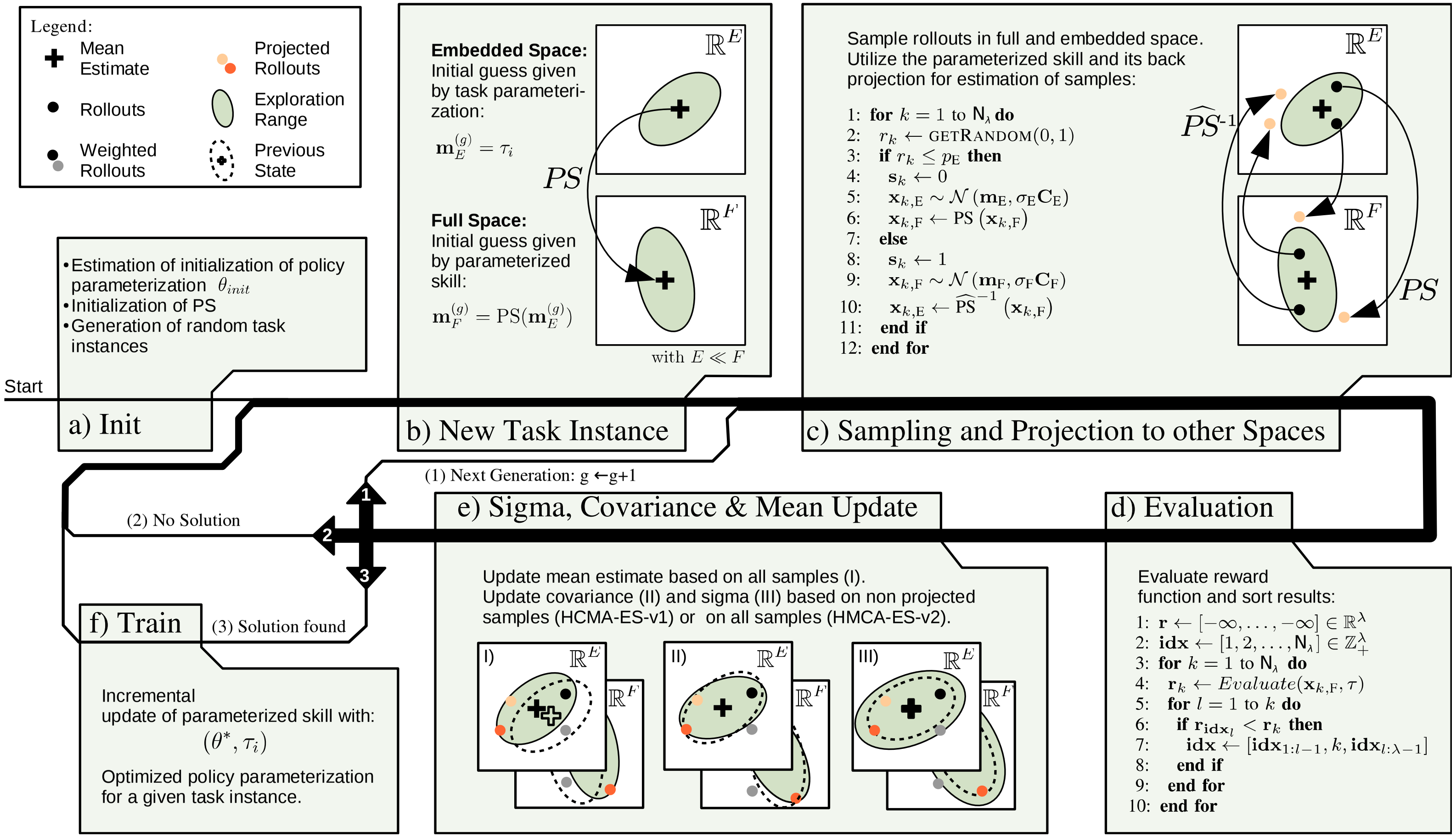
4.2.1 Component Selection

The following presents a brief introduction of the chosen policy representation and the algorithm for policy optimization and learning that are used throughout this chapter. The component selection is closely related to the previously presented bootstrapping experiments in Section 3.2.1.

a) **Selection of Policy Representation:**
The proposed method does not rely on a specific type of policy representation. Many methods for compact policy presentation have been proposed, e.g. based on Gaussian Mixture Regression (GMR) [Günter et al., 2007] or Neural Imprinted Vector Fields [Lemme et al., 2014], as discussed in Section 2.2.2. This chapter refers to Dynamic Motion Primitives (DMP, [Ijspeert et al., 2013]), in the same configuration as motivated in Section 3.2.1.

b) **Selection of Policy Optimization Algorithm:**
For optimization of DMP parameters $\boldsymbol{\theta}^*$ given a task $\boldsymbol{\tau}$, the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES, [Hansen, 2006]) is applied, as introduced in Section 2.2.2. Stulp and Sigaud [2013] have shown that the black-box optimization by CMA-ES is very efficient and reliable in combination with DMPs. Besides an optimization in one parameter space, this chapter will propose an extension of CMA-ES to perform an optimization in multiple spaces. Detailed information regarding CMA-ES is presented in Section 2.2.2.

c) **Selection of Learning Algorithm:**
To allow the comparison of the methods proposed in this chapter to the bootstrapping of parameterized skills as presented in Chapter 3, the learner configuration was kept unchanged. For learning of parameterized skills $\mathrm{PS}(\boldsymbol{\tau})$ an incremental variant of the Extreme Learning Machine (ELM, [Huang et al., 2006]) was implemented as discussed in Section 3.2.1. As before, hidden Layer size was set to $N_{\mathrm{H}} = 50$ for generalization in joint space and $N_{\mathrm{H}} = 20$ in case of a cartesian end effector space. Linear regression is applied on a random projection of the input $\mathbf{W}^{inp} \in \mathbb{R}^{N_{\mathrm{H}} \times E}$, a nonlinear transformation $\sigma(x) = (1+e^{-x})^{-1}$ and a linear output transformation $\mathbf{W}^{out} \in \mathbb{R}^{F \times N_{\mathrm{H}}}$ that can be updated by incremental least squares algorithms. A more detailed discussion on the learning method and parameter estimation of the readout weights is presented in Section 2.2.2.



- Estimation of initialization of policy parameterization θ_{init}
- Initialization of PS
- Generation of random task instances

Embedded Space: Initial guess given by task parameterization:

$\mathbf{m}_E^{(g)} = \tau_i$

Full Space: Initial guess given by parameterized skill:

$\mathbf{m}_F^{(g)} = \mathrm{PS}(\mathbf{m}_E^{(g)})$

Sample rollouts in full and embedded space. Utilize the parameterized skill and its back projection for estimation of samples:

```
for $k = 1$ to $\mathrm{N}_\lambda$ do
  $r_k \leftarrow \mathrm{GETRANDOM}(0,1)$
  if $r_k \leq p_\mathrm{E}$ then
    $\mathbf{s}_k \leftarrow 0$
    $\mathbf{x}_{k,\mathrm{E}} \sim \mathcal{N}(\mathbf{m}_\mathrm{E}, \sigma_\mathrm{E}\mathbf{C}_\mathrm{E})$
    $\mathbf{x}_{k,\mathrm{F}} \leftarrow \mathrm{PS}\left(\mathbf{x}_{k,\mathrm{F}}\right)$
  else
    $\mathbf{s}_k \leftarrow 1$
    $\mathbf{x}_{k,\mathrm{F}} \sim \mathcal{N}(\mathbf{m}_\mathrm{F}, \sigma_\mathrm{F}\mathbf{C}_\mathrm{F})$
    $\mathbf{x}_{k,\mathrm{E}} \leftarrow \widehat{\mathrm{PS}}^{-1}\left(\mathbf{x}_{k,\mathrm{F}}\right)$
  end if
end for
```

Evaluate reward function and sort results:

```
$\mathbf{r} \leftarrow [-\infty, \ldots, -\infty] \in \mathbb{R}^\lambda$
$\mathbf{idx} \leftarrow [1, 2, \ldots, \mathrm{N}_\lambda] \in \mathbb{Z}_+^\lambda$
for $k = 1$ to $\mathrm{N}_\lambda$ do
  $\mathbf{r}_k \leftarrow Evaluate(\mathbf{x}_{k,\mathrm{F}}, \tau)$
  for $l = 1$ to $k$ do
    if $\mathbf{r}_{\mathbf{idx}_l} < \mathbf{r}_k$ then
      $\mathbf{idx} \leftarrow [\mathbf{idx}_{1:l-1}, k, \mathbf{idx}_{l:\lambda-1}]$
    end if
  end for
end for
```

Update mean estimate based on all samples (I). Update covariance (II) and sigma (III) based on non projected samples (HCMA-ES-v1) or on all samples (HMCA-ES-v2).

Incremental update of parameterized skill with:

(θ^*, τ_i)

Optimized policy parameterization for a given task instance.

Figure 4.3: Proposed optimization loop for the bootstrapping of parameterized skills in hybrid spaces. After initialization (a), optimization for a new task instance is initiated (b). Optimization is performed (1) until stopping criterion is reached and no solution was found (2) or the optimized solution fulfills the task (3). Update of CMA-ES (I-III) is performed simultaneously for the task and policy space.

4.2.2 CMA-ES in Hybrid Spaces

The implementation of the proposed hybrid optimization method is based on CMA-ES [Hansen, 2006]. The original algorithm of CMA-ES relies on four main steps, detailed information can be found in Section 2.2.2. Optimization is performed in generations, which means that an update of the mean estimate is performed based on observation of rewards from actions that have to be performed under several perturbations. CMA-ES has an internal representation of the current mean and of the covariance matrix that allows for sampling of normally distributed actions around the current mean. In addition, CMA-ES estimates an evolution path for the mean and the covariance matrix update. Those evolution paths allow for more stability to outliers and noise. The first step performs the sampling from a multivariate normal distribution centered at the current estimate by Equation 2.28. Followed by the update of the estimated solution for the next generation with respect to the rewards of the sampled rollouts by Equation 2.29. The third step targets the update of the covariance matrix and its evolution path, given by Equation 2.31 and Equation 2.30. And the final step performs an update of the exploration width and its assigned evolution path as in Equation 2.33 and in Equation 2.32.

To be able to perform CMA-ES in hybrid spaces, the CMA-ES algorithm is applied on two parameter spaces simultaneously. Added indices F and E indicate the affiliation of variables for optimization to policy space (F) and task space (E). Two distinct means $\mathbf{m}_{\mathrm{E}}^{(g+1)}$ and $\mathbf{m}_{\mathrm{F}}^{(g+1)}$ represent the current optimum to minimize the objective function, i.e. negative reward. Covariance matrices $\mathbf{C}_{\mathrm{E}}^{(g+1)}$ and $\mathbf{C}_{\mathrm{F}}^{(g+1)}$ as well as their evolution paths $\mathbf{p}_{c,\mathrm{E}}^{(g+1)}$ and $\mathbf{p}_{c,\mathrm{F}}^{(g+1)}$ allow for random normal distributed perturbation of the respective mean. The variances $\sigma_{\mathrm{E}}^{(g+1)}$ and $\sigma_{\mathrm{F}}^{(g+1)}$ in addition to their evolution paths $\mathbf{p}_{\sigma,\mathrm{E}}^{(g+1)}$ and $\mathbf{p}_{\sigma,\mathrm{F}}^{(g+1)}$ define the exploration size in each space. In comparison to two independent CMA-ES optimizations in each space, probabilities $p_{\mathrm{E}}^{(g+1)}$ respectively $p_{\mathrm{F}}^{(g+1)} = 1 - p_{\mathrm{E}}^{(g+1)}$ are introduced that indicate in which space sampling of the rollouts is performed. $p_{\mathrm{E}}^{(g+1)}$ and $p_{\mathrm{F}}^{(g+1)}$ can be interpreted as mixing coefficients that allow for interpolation between a CMA-ES optimization in the task space ($p_{\mathrm{E}}^{(g+1)} = 1$) and a CMA-ES optimization in policy space ($p_{\mathrm{E}}^{(g+1)} = 0$). For each update step of generation $(g+1)$, a combined update based on $k = 1, .., \lambda_{\mathrm{H}}^{(g+1)}$ samples is performed. Each sample is annotated by $\mathbf{s}_k^{(g+1)} = 0$ if the rollout k was sampled in the task space or by $\mathbf{s}_k^{(g+1)} = 1$ if it was sampled in the policy space. The initialization (Figure 4.3(b)) for a new task instance i of the parameterization in the embedded space is $\mathbf{m}_{\mathrm{E}}^{(g=0)} = \boldsymbol{\tau}$ and the initialization in full space is given by the generalization of the parameterized skill $\mathbf{m}_{\mathrm{F}}^{(g+1)} = \boldsymbol{\theta}_{\mathrm{PS}} = \mathrm{PS}(\mathbf{m}_{\mathrm{E}}^{(g=0)})$. The sampling of rollouts is given by

$$\begin{aligned} \mathbf{x}_{k,\mathrm{E}}^{(g+1)} &= \mathbf{m}_{\mathrm{E}}^{(g)} + \sigma_{\mathrm{E}}^{(g)} \mathbf{y}_{k,\mathrm{E}}^{(g+1)} \qquad \text{for } k = 1, .., .\lambda_H \wedge \mathbf{s}_k^{(g+1)} = 0 \qquad \text{and} \\ \mathbf{x}_{k,\mathrm{F}}^{(g+1)} &= \mathbf{m}_{\mathrm{F}}^{(g)} + \sigma_{\mathrm{F}}^{(g)} \mathbf{y}_{k,\mathrm{F}}^{(g+1)} \qquad \text{for } k = 1, .., .\lambda_H \wedge \mathbf{s}_k^{(g+1)} = 1. \end{aligned} \tag{4.2}$$

For each rollout the selection of the target space for sampling is based on probabilities $p_{\mathrm{E}}^{(g+1)}$ and $p_{\mathrm{F}}^{(g+1)}$. Rollouts are draw from a normal distribution defined as

$$\mathbf{y}_{k,\mathrm{E}}^{(g+1)} =\sim \mathcal{N}_k(\mathbf{0}, \mathbf{C}_{\mathrm{E}}^{(g+1)}) \quad \text{and} \quad \mathbf{y}_{k,\mathrm{F}}^{(g+1)} =\sim \mathcal{N}_k(\mathbf{0}, \mathbf{C}_{\mathrm{F}}^{(g+1)}). \tag{4.3}$$

The number of evaluated rollouts per generation is defined as a mixture of λ_{E} and λ_{F}, given by

$$\lambda_{\mathrm{H}}^{(g+1)} = p_{\mathrm{E}}^{(g+1)}\lambda_{\mathrm{E}} + p_{\mathrm{F}}^{(g+1)}\lambda_{\mathrm{F}} = 4 + p_{\mathrm{E}}^{(g+1)} \lfloor 3 \ln E \rfloor + p_{\mathrm{F}}^{(g+1)} \lfloor 3 \ln F \rfloor . \tag{4.4}$$

The update of λ_{H} is motivated by the number of rollouts per generation $\lambda = \lfloor 3 \ln D \rfloor$ in relation to the dimensionality D of the optimization problem as introduced for CMA-ES ([Hansen, 2006]). In a next step, the parameterized skill performs a mapping of samples originated in task space to policy space and vise versa (see Figure 4.3(c), line 6 and 10). Therefore, a parameterized skill is required that allows for inverse evaluation notated as $\widehat{\mathrm{PS}}^{-1}$. This mapping process is defined as

$$\begin{aligned} \mathbf{x}_{k,\mathrm{E}}^{(g+1)} &= \widehat{\mathrm{PS}}^{(g)^{-1}}(\mathbf{x}_{k,\mathrm{F}}^{(g+1)}) \qquad \text{for } k = 1, .., .\lambda_H \wedge \mathbf{s}_k^{(g+1)} = 1 \quad \text{and} \\ \mathbf{x}_{k,\mathrm{F}}^{(g+1)} &= \widehat{\mathrm{PS}}^{(g)^{-1}}(\mathbf{x}_{k,\mathrm{F}}^{(g+1)}) \qquad \text{for } k = 1, .., .\lambda_H \wedge \mathbf{s}_k^{(g+1)} = 0. \end{aligned} \tag{4.5}$$

A representation of all rollouts in $\mathbf{x}_{k,\mathrm{F}}^{(g+1)}$ allows for execution of the policy and evaluation of the reward function. The rollouts are ordered based on the magnitude of the respective reward as proposed by the original CMA-ES approach [Hansen, 2006], as shown in Figure 4.3d. At this point an update of the means $\mathbf{m}_{\mathrm{E}}^{(g+1)}$ and $\mathbf{m}_{\mathrm{F}}^{(g+1)}$ with respect to $\mathbf{x}_{k,\mathrm{E}}^{(g+1)}$ and $\mathbf{x}_{k,\mathrm{F}}^{(g+1)}$ by applying Equation 2.29 is possible. This allows for an update of the estimated means in both spaces based on all rollouts that have been evaluated in the current generation. Note, that the means $\mathbf{x}_k^{(g+1)}$ do not develop independently. Rather they are linked by the mapping of the parameterized skill. The success rate of both spaces results in an adaptation of the mixture of rollouts that are performed in the policy and task space (p_{F} and p_{E}). The success rate is defined by the ratio of successful rollouts (rollouts that exceed the current reward maximum), encoded by the weights $\mathbf{w}_k^{(g+1)}$ as well as space that was used for sampling $\mathbf{s}_k^{(g+1)}$ of the performed rollouts. The update of p_{E} as well as p_{F} is given by

$$\delta p_{\mathrm{E}}^{(g+1)} = \frac{\sum^{\mu}_{\substack{k=1, \\ \mathbf{s}_k^{(g+1)}=0}} \mathbf{w}_k^{(g+1)}}{\sum_{k=1}^{\mu} \mathbf{w}_k^{(g+1)}} - \delta p_{\mathrm{E}}^{(g)}, \quad \delta p_{\mathrm{F}}^{(g+1)} = -\delta p_{\mathrm{E}}^{(g+1)}. \tag{4.6}$$

For experimental evaluation, two approaches for an update of the covariance and exploration width are exploited: The first version utilizes only samples that originate in the same space for an update of the covariance $\mathbf{C}$ and exploration width σ; The second version utilizes the mapping of PS and $\widehat{\mathrm{PS}}^{-1}$ to estimate an additional update of the covariance and the exploration width with respect to all samples. This work will refer to the first version as **H**ybrid **C**ovariance **M**atrix **A**daptation - **E**volutionary **S**trategy - **V**ersion 1 (*HCMA-ES-v1*) and to the second version as *HCMA-ES-v2*.

HCMA-ES-v1 The update of the covariance, exploration radius and their evolution paths is performed as in the original CMA-ES algorithm, depicted in Equation 2.31 to Equation 2.32. The update step for each space, encoded in $\mathbf{s}_k^{(g+1)}$, considers only rollouts sampled in the same space. The normalization of $\sum \mathbf{w}_k^{(g+1)} = 1$ for all $\mathbf{s}_k^{(g+a)} = 0$ in case of the task space as well as $\mathbf{s}_k^{(g+a)} = 1$ in case of the policy space is necessary since not all samples are used in each space. Additionally, the estimation of $\mu_{\text{eff,E}}$ and $\mu_{\text{eff,F}}$ with respect to $\mathbf{s}$ has to be performed as well. The neglection of projected samples for the update of the covariance and its exploration width allows for simplification of the combination process. But this simplification prevents that e.g. the covariance in the high-dimensional policy space can shape into the direction of samples along the low-dimensional manifold of the task parameterization.

HCMA-ES-v2 The parameterized skill can be regarded as a mapping between the high-dimensional policy parameterization and a low-dimensional embedding. The mapping process of parameterizations sampled from multivariate normal distributions to other spaces results in distorted distributions in the target space due to the nonlinear transformation of the parameterized skill. For the integration of projected samples in the update of the covariance, a rescaling of projected samples is necessary to cancel out the effect of the exploration width σ. The update of the exploration width σ requires the estimation of the distribution of projected samples, but the estimation of a covariance of projected rollouts requires a large number of samples which is not feasible for the presented scenarios (≈ 10 rollouts per generation). An update of exploration width $\sigma_{\text{E}}^{(g+1)}$ and $\sigma_{\text{F}}^{(g+1)}$ with respect to each other is performed by the estimation of a scaling factor between the evaluated rollouts of the current generation, which allows the application for low sample numbers.

To consider samples from other spaces for an update of the covariance and its evolution path, samples from other spaces have to be rescaled to keep the covariance $\mathbf{C}$ at a constant size, i.e. $\det\left(\mathbf{C}^T\mathbf{C}\right) = \prod \lambda_i = \textit{const.}$, the product of eigenvalues of $\mathbf{C}$ is constant. From the update of the exploration width of CME-ES as given in Equation 2.32 and Equation 2.33, the condition $\sqrt{\mu_{\text{eff}}}\mathbf{C}^{(g)^{-1/2}}\mathbf{y}_{\mathbf{w}} = \text{E}\left|\left|\mathcal{N}\left(\mathbf{0},\mathbf{I}\right)\right|\right|$ for constant covariance size can be inferred. Therefore, an appropriate scaling factor to the calculation of the weighted sum is added, resulting in a modified estimation of $\tilde{\mathbf{y}}_{\mathbf{w},\text{E}}$ with an additional scaling of samples that originate in the full space. As given by

$$\begin{aligned}\tilde{\mathbf{y}}_{\mathbf{w},\mathrm{E}} &= \sum_{\substack{i=1\\ \mathbf{s}_{i:\lambda}^{(g+1)}=0}}^{\mu} \mathbf{w}_i \mathbf{y}_{i:\lambda,E}^{(g+1)} + \frac{\chi_E}{\beta_\mathrm{E}} \sum_{\substack{i=1\\ \mathbf{s}_{i:\lambda}^{(g+1)}=1}}^{\mu} \mathbf{w}_i \mathbf{y}_{i:\lambda,E}^{(g+1)}, \quad \text{with} \\ \beta_\mathrm{E} &= \sqrt{\mu_{eff}} \left\| \mathbf{C}_\mathrm{E}^{-1/2} \sum_{\substack{i=1\\ \mathbf{s}_{i:\lambda}^{(g+1)}=1}}^{\mu} \frac{\mathbf{w}_i \mathbf{y}_{i:\lambda,E}^{(g+1)}}{\alpha_\mathrm{F}} \right\|, \quad \alpha_\mathrm{F} = \sum_{\substack{j=1\\ \mathbf{s}_{j:\lambda}^{(g+1)}=1}}^{\mu} \mathbf{w}_j.\end{aligned} \tag{4.7}$$

$\chi_N \overset{\text{def}}{=} \sqrt{\mathrm{E}(\chi_N^2)} = \mathrm{E}\,||\mathcal{N}(\mathbf{0}, I_N)||$ refers to the chi-squared distribution χ_N^2 with N degrees of freedom. The estimation of $\tilde{\mathbf{y}}_{\mathbf{w},\mathrm{F}}$ is performed likewise, as

$$\begin{aligned}\tilde{\mathbf{y}}_{\mathbf{w},\mathrm{F}} &= \sum_{\substack{i=1\\ \mathbf{s}_{i:\lambda}^{(g+1)}=1}}^{\mu} \mathbf{w}_i \mathbf{y}_{i:\lambda,F}^{(g+1)} + \frac{\chi_N}{\beta_\mathrm{F}} \sum_{\substack{i=1\\ \mathbf{s}_{i:\lambda}^{(g+1)}=0}}^{\mu} \mathbf{w}_i \mathbf{y}_{i:\lambda,F}^{(g+1)}, \quad \text{with} \\ \beta_\mathrm{F} &= \sqrt{\mu_{eff}} \left\| \mathbf{C}_\mathrm{F}^{-1/2} \sum_{\substack{i=1\\ \mathbf{s}_{i:\lambda}^{(g+1)}=0}}^{\mu} \frac{\mathbf{w}_i \mathbf{y}_{i:\lambda,F}^{(g+1)}}{\alpha_E} \right\|, \quad \alpha_E = \sum_{\substack{j=1\\ \mathbf{s}_{j:\lambda}^{(g+1)}=0}}^{\mu} \mathbf{w}_j.\end{aligned} \tag{4.8}$$

The factor β_E^{-1} results in a rescaling of samples from policy parameter space to task space and β_F^{-1} scales from task parameter space to policy space. The update of $\mathbf{p}_c$ and $\mathbf{C}$ can be achieved by Equation 2.31 and Equation 2.30 with respect to $\tilde{\mathbf{y}}_{\mathbf{w},\mathrm{E}}$ and $\tilde{\mathbf{y}}_{\mathbf{w},\mathrm{F}}$. The final step updates the exploration width $\sigma_\mathrm{E}^{(g+1)}$ and $\sigma_\mathrm{F}^{(g+1)}$. It is achieved by performing a mixing of the updated sigma of the own space and the rescaled sigma of the second space based on the success rate of the spaces, as

$$\begin{aligned}\sigma_\mathrm{E}^{(g+1)} &= p_\mathrm{E}\tilde{\sigma}_\mathrm{E}^{(g+1)} + p_\mathrm{F}\frac{\tilde{\sigma}_\mathrm{F}^{(g+1)}\beta_E}{\chi_E} \quad \text{and} \\ \sigma_\mathrm{F}^{(g+1)} &= p_\mathrm{F}\tilde{\sigma}_\mathrm{F}^{(g+1)} + p_\mathrm{E}\frac{\tilde{\sigma}_\mathrm{E}^{(g+1)}\beta_F}{\chi_F}.\end{aligned} \tag{4.9}$$

The evaluation of the properties of both algorithm versions and a comparison to classical CMA-ES will be presented in the successive experiments, Section 4.2.4 and Section 4.2.5.

4.2.3 Implementation of the Parameterized Skill

The proposed optimization method does not rely on a specific learning method. But in comparison to the bootstrapping of the parameterized skill as proposed in

Section 3.2, the policy search in hybrid spaces requires an inverse estimate of the parameterized skill. Therefore, the learner must be continuous and locally differentiable. Further candidates for this task are associative memories due to their bidirectional representation of inputs and outputs. A more in-depth discussion on associative memories can be found in Section 2.2.2. For the implementation that is presented in the following a different approach is used, the Jacobian of the parameterized skill is used to iteratively estimate a proper input $\boldsymbol{\tau}$ for a required output $\boldsymbol{\theta}$. Since the implementation of the parameterized skill (used for the bootstrapping experiments in Chapter 3) is kept unchanged, a comparison of both methods under equal conditions is possible. The local inverse estimate of the parameterized skill is motivated by the Inverse Function Theorem by [Spivak, 1971], that states that it is possible to estimate a local inverse of a function if the determinant of the Jacobian is not zero. The estimation of the change in the policy parameter space that is caused by a change in the task space is given by

$$\Delta\boldsymbol{\theta}^{*} \approx J_{\mathrm{PS}}(\boldsymbol{\tau}^{*})\Delta\boldsymbol{\tau}^{*}. \tag{4.10}$$

Since the parameterized skill is not a bijective mapping, multiple solutions can exist. Optimization is assumed to sample in a local neighbourhood of the current estimate, therefore, the gradient descent is initialized with $\mathrm{PS}(\boldsymbol{\tau})$. Gradient descent is implemented by the Levenberg-Marquardt method [Liu and Han, 2003], also referred to as Damped Least-Squares method as depicted e.g. in [Buss, 2004], due to numerical stability in comparison to pseudoinverse and Jacobian transposed based methods. The incremental update of the estimated task space $\boldsymbol{\tau}^{*}$ is based on the Jaocbian $J_{\mathrm{PS}}(\boldsymbol{\tau}^{*})$ of PS with respect to the input $\boldsymbol{\tau}^{*}$

$$\begin{aligned}\Delta\boldsymbol{\tau}^{*} = J_{\mathrm{PS}}(\boldsymbol{\tau}^{*})^{\top}\left(J_{\mathrm{PS}}(\boldsymbol{\tau}^{*})J_{\mathrm{PS}}(\boldsymbol{\tau}^{*})^{\top} + \lambda^{2}I\right)^{-1}\mathbf{e},\\ \text{with } \mathbf{e} = \left(\boldsymbol{\theta}^{*} - \mathrm{PS}(\boldsymbol{\tau}^{*})\right).\end{aligned} \tag{4.11}$$

4.2.4 Evaluation on Toy Data

To gain insight into the proposed hybrid search method, experimental investigation on four test cases is performed. For simplicity and visualization purposes, the policy is defined as the identity $\pi_{\boldsymbol{\theta}} = \boldsymbol{\theta}$ and the reward function operates directly on $\boldsymbol{\theta} \in \mathbb{R}^{2}$, the 2D space of the policy parameterization. The reward function is parameterized by $\tau \in \mathbb{R}^{1}$ defining the position of maximum reward in the 2D space. This allows to visualize the reward function in relation to a fixed value of τ. A visualization of both reward functions for several fixed parameterizations τ are shown in Figure 4.4. The color intensity encodes the reward for a given task parameterization $\boldsymbol{\theta}$. The first scenario describes a circular manifold with a maximum at $\mathbf{m}_{\tau}$, where the reward is

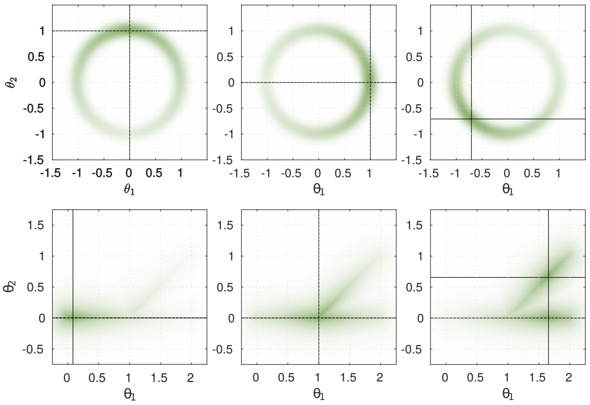


Figure 4.4: Visualization of the designed reward functions. Circular reward function R_{circular} (top) and branch reward R_{branch} (bottom) for three different task parameterizations are shown. Crossing points of horizontal and vertical black lines indicate maxima of reward functions. For $\tau > 1$ multiple maxima of the reward function exist (bottom-right). Color intensity indicates the magnitude of the reward for a depicted parameterization $\boldsymbol{\theta}$

given by

$$\begin{aligned} R_{\text{a}}(\boldsymbol{\theta}, \tau) &= \frac{1}{\sqrt{2\pi\sigma_{\text{a}}^2}} \exp - \frac{\left|\text{atan2}\left(\mathbf{m}_\tau \times \boldsymbol{\theta}, \mathbf{m}_\tau \cdot \boldsymbol{\theta}\right)\right|}{2\sigma_{\text{a}}^2} \quad \text{and} \\ R_{\text{r}}(\boldsymbol{\theta}) &= \frac{1}{\sqrt{2\pi\sigma_{\text{r}}^2}} \exp - \frac{(1 - \|\boldsymbol{\theta}\|)^2}{2\sigma_{\text{r}}^2}, \text{ with } \mathbf{m}_\tau = \begin{bmatrix} \sin(\tau) \\ \cos(\tau) \end{bmatrix}. \end{aligned} \tag{4.12}$$

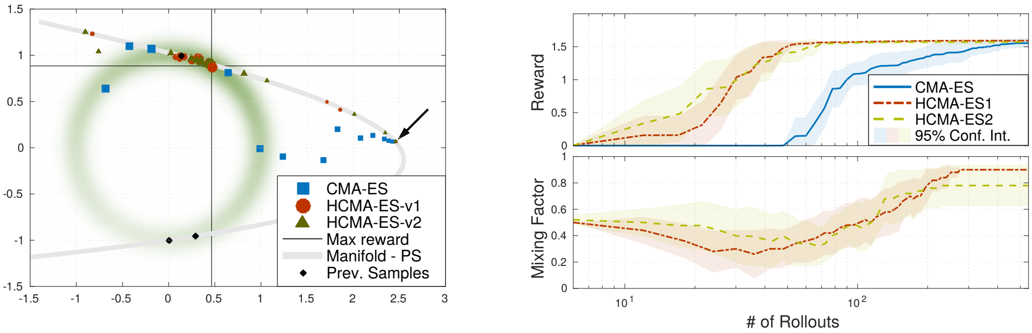
The reward function includes the angular deviation $R_{\text{a}}(\boldsymbol{\theta})$ as well as the deviation in the radius $R_{\text{r}}(\boldsymbol{\theta})$, which are weighted by Gaussian functions. The overall reward is given by $R_{\text{circular}}(\boldsymbol{\theta}, \tau) = R_{\text{a}}(\boldsymbol{\theta}, \boldsymbol{\tau}) \cdot R_{\text{r}}(\boldsymbol{\theta})$.

The second reward function is based on a branch manifold. For parameterizations $\tau \leq 1$ the maximum reward is located at $[\tau; 0]$. For $\tau > 1$ two maxima can be found at $[\tau; 0]$ and $[\tau; 1{+}\tau]$. It is based on a combination of the distances to the parameterized maxima of the function $R_{\text{m}}(\boldsymbol{\theta}, \tau)$ and the distance to the branch manifold R_{b} that is defined as

$$\begin{aligned} R_{\text{m}}(\boldsymbol{\theta}, \tau) &= \frac{1}{\sqrt{2\pi\sigma_{\text{d}}^2}} \exp - \frac{d_{\text{min}}}{2\sigma_{\text{d}}^2}, \text{ with} \\ d_{\text{min}} &= \begin{cases} ||\boldsymbol{\theta} - [\tau; 0]\,||, & \text{if } \tau \leq 1 \\ \min(||\boldsymbol{\theta} - [\tau; 0]\,||, ||\boldsymbol{\theta} - [\tau; \tau - 1]\,||), & \text{else} \end{cases} \\ \text{and} \quad R_{\text{b}}(\boldsymbol{\theta}) &= \frac{1}{\sqrt{2\pi\sigma_{\text{r}}^2}} \exp - \frac{Dist_{\text{branch}}(\boldsymbol{\theta})}{2\sigma_{\text{r}}^2}. \end{aligned} \tag{4.13}$$

With $Dist_{\text{branch}}(\boldsymbol{\theta})$, the minimum distance of $\boldsymbol{\theta}$ to the line segments $[0; 0] - [2; 0]$ and $[1; 0] - [2; 1]$. The combination of both reward terms results in the final reward function $R_{\text{branch}}(\boldsymbol{\theta}, \tau) = R_{\text{m}}(\boldsymbol{\theta}, \boldsymbol{\tau}) \cdot R_{\text{b}}(\boldsymbol{\theta})$. The scenario is designed in such a way that it reflects expected real world problems: The space of all possible actions includes a subset of appropriate actions on a manifold that have higher rewards. Within this subset, a maximum of the reward function is expected at parameterizations that solve the task in an appropriate way. The four evaluated test cases are shown in Figure 4.5 to Figure 4.8. Each plot shows the comparison between a search in the policy space by CMA-ES as well as the behavior of the proposed hybrid algorithms. The green color intensity encodes the reward for a depicted policy parameterization $\boldsymbol{\theta}$. Previous training data (obtained by estimation of one maximum of the reward function) of the parameterized skill is indicated by a black dot (◆). Based on the training data, the mapping f_{PS} on the manifold in the policy space, i.e. $\text{PS}(\tau)$, is constructed and shown as a gray line. The symbols ■,● and ▲ represent the current estimates of the means $\mathbf{m}_F^g$ in the policy space, whereas the size ■-▪,●-• and ▲-▴ indicates the history of previous mean estimates $\mathbf{m}_F^{g-n}$, $\forall n \in \{1, \ldots, g\}$, up to the first generation, with a decreasing size of the symbol. The real maximum of the reward function is marked by black crossing lines and the location of the initial estimate $\theta_{\text{PS}} = \text{PS}(\tau_i)$ on f_{PS} is highlighted by a black arrow (↙).

In the following all four scenarios (a-d) and the optimization process will be presented in detail:

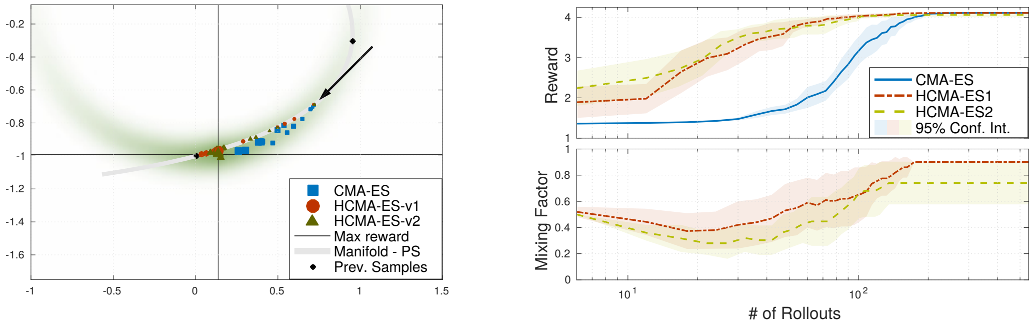


(a) (b)

Figure 4.5: Comparison of optimization algorithms on 2D reward function: Overshoot of PS, hybrid optimization is able to utilize manifold of the parameterized skill (gray line) to perform optimization in 1D space. (a) Estimated means during optimization, marker size indicates the generation. Black arrow points to initial guess on manifold (gray line) of parameterized skill. (b) The comparison of reward and mixing factor during optimization is shown.

Overshoot Scenario The scenario in Figure 4.5a shows a situation in which an overshoot of the estimation of the parameterized skill occurs. This scenario utilizes the circular reward R_{circular} and performs an exponential distortion $f_{\text{dist}}(\tau) = \exp(\tau) * \pi / \exp(\pi)$ of the parameterization to enforce a faulty generalization of the memory resulting in $R(\boldsymbol{\theta}, \tau) = R_{\text{circular}}(\boldsymbol{\theta}, f_{\text{dist}}(\tau))$. For training of the parameterized skill, the optimal parameterizations $\boldsymbol{\theta}_i$ for three different task instances τ_i are estimated.

It can be seen that for the depicted task parameterization, the parameterized skill proposes a solution that is located in a region with a little gradient information. By following the low-dimensional embedding of the parameterized skill, the hybrid approach is able to guide the optimizer into a region with a stronger gradient and that is closer to a desired maximum of the reward function. For the original CMA-ES approach it takes longer to reach a region with more informative gradient information and requires therefore more rollouts in comparison to the hybrid optimization in both spaces. Algorithm HCMA-ES-v1 and HCMA-ES-v2 show a comparable performance and a similar behavior during optimization. Investigations of the shape of covariance reveal the extended update policy of HCMA-ES-v2. Since the shape and size of the covariance of the policy space integrates rollouts sampled in the task space as well, the covariance grows and shapes aggressively into the direction of the real maximum and the shape of the manifold of the parameterized skill. In a region close to the maximum, i.e. the covariance shrinks but keeps the shape influenced by the previous fast approaching phase in the low-dimensional manifold. Figure 4.5b shows the probability of performing a rollout in the policy parameter space, starting at equal probabilities for both spaces, the algorithm first shifts its focus to the task space and switches to a fine-tuning at the end of the optimization phase.



(a) (b)

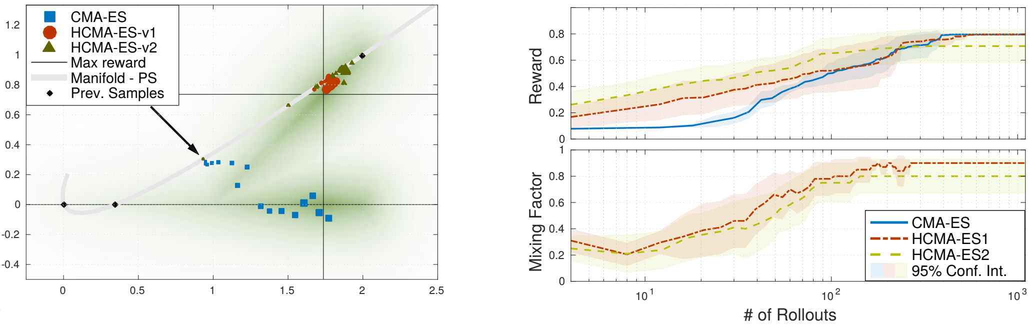
Figure 4.6: Comparison of optimization algorithms on 2D reward function: Distorted estimates of the parameterized skill. (a) Estimated means during optimization, marker size indicates the generation. Black arrow points to initial guess on manifold (gray line) of parameterized skill. (b) The comparison of reward and mixing factor during optimization is shown.

Distortion Scenario In case of a distortion of the parameterized skill, it is assumed that a sufficient solution can be found in the manifold f_{PS}, e.g. $\boldsymbol{\theta}^* = \text{PS}(\boldsymbol{\tau} + \boldsymbol{\epsilon})$. An exemplary evaluation of this situation can be seen in Figure 4.6.

The experiment is performed for the circular reward R_{circular} as in the previous example but a sigmoidal distortion of the parameterization $f_{\text{dist}}(\tau) = \pi/(1+\exp(-4*$

$(\tau - \pi/2)))$ is applied to enforce a distorted estimate of the parameterized skill, resulting in $R(\boldsymbol{\theta}, \tau) = R_{\text{circular}}(\boldsymbol{\theta}, f_{\text{dist}}(\tau))$. Again, three optimal parameterizations $\boldsymbol{\theta}_i$ for three different task instances τ_i are presented to the memory as training samples.

During optimization, the hybrid search incorporates a lower dimensional search space and is able to follow the gradient of the reward function in the task space. Due to the shape of the designed reward function, the standard CMA-ES approach (that operates in policy space) has to adapt its covariance along the circular structure of the reward function and approaches slower to the position of the maximum, as shown in Figure 4.6a. The optimization in hybrid spaces benefits from the projection onto a manifold f_{PS} that compensates for the circular structure and allows for a fast convergence. Figure 4.6b shows the comparison of the reached reward in relation to performed rollouts. As discussed for the overshoot scenario, in case of HCMA-ES-v2 a shaping of the covariance of the policy space can be observed, that is influenced by the rollouts along the manifold of the parameterized skill.



br

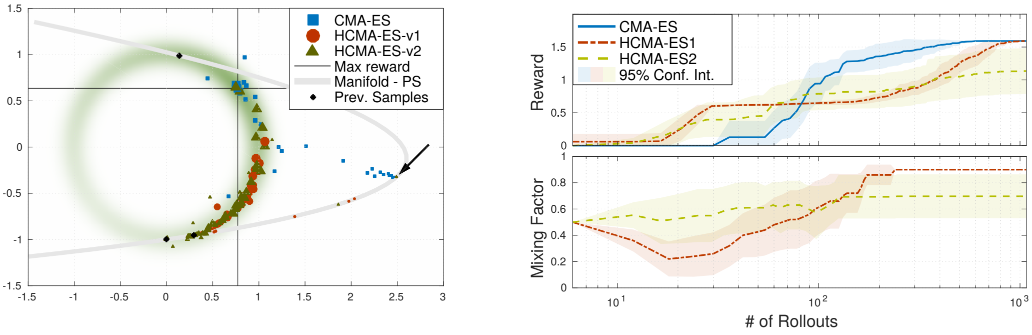
(a)

(b)

Figure 4.7: Comparison of optimization algorithms on 2D reward function: Multiple maxima of reward function. (a) Estimated means during optimization, marker size indicates the generation. Black arrow points to initial guess on manifold (gray line) of the parameterized skill. (b) The comparison of reward and mixing factor during optimization is shown.

Multiple Minima Scenario This scenario explores tasks with multiple solutions for a certain range of task parameterizations. Evaluation is performed on the branch reward R_{branch} in combination with the exponential distortion $f_{\text{dist}}(\tau) = \exp(\tau) * \pi/\exp(\pi)$ used in the overshoot scenario. Therefore, the reward function is given by $R(\boldsymbol{\theta}, \tau) = R_{\text{branch}}(\boldsymbol{\theta}, f_{\text{dist}}(\tau))$. The presented training samples for the parameterized skill as well as the experimental setup can be seen in Figure 4.7a. As discussed in Section 4.2, multiple maxima of the reward function bear the risk of generating inconsistent training data for the parameterized skill and impede generalization capabilities. It is beneficial to prefer solutions for tasks that are close to the manifold of the parameterized skill, as in this scenario, solutions on the upper branch of the reward function, as shown in Figure 4.7a. A hybrid optimization that performs a search along the manifold of the parameterized skill enhances the probability to find an optimum close to the manifold of the already established parameterized skill. As shown in Figure 4.7a, starting from the initial guess, the standard CMA-ES approach follows the gradient towards the manifold of the reward function. The covariance, responsible for perturbation of sampling, starts to shape into that direction and causes the optimizer to follow the gradient towards the lower branch of the reward function. The estimated solution is far off the manifold of the parameterized skill and results in inconsistent training data since previous training data was selected from the upper branch. The standard CMA-ES optimization is able to find a solution for the given task without requiring a significantly different number of rollouts than the hybrid optimization methods.

Although the hybrid search cannot speed up the optimization process, the optimizer relies on the manifold of the parameterized skill to move towards the gradient of the reward function, as shown in Figure 4.7b. For the final phase, the optimizer switches the preference to the policy parameter space for optimization. It is able to find a maximum of the reward function that is consistent with previous training data since it is located in the upper branch of the manifold of the reward function.



(a) (b)

Figure 4.8: Comparison of optimization algorithms on 2D reward function: Overshoot of parameterized skill, standard CMA-ES is able to perform a more efficient optimization than hybrid methods. (a) Estimated means during optimization, marker size indicates the generation. Black arrow points to initial guess on manifold (gray line) of parameterized skill. (b) The comparison of reward and mixing factor during optimization is shown.

Failed Overshoot Scenario In case the estimate of the parameterized skill is of very low quality, optimization in the low-dimensional space of f_{PS} can lead to a fast convergence to a region with higher rewards (as in Figure 4.5). But the algorithm could end up at a parameterization that is far away from the desired solution, so that an optimization by CMA-ES can reach a high reward with less number of executed rollouts. This situation is shown in Figure 4.8, it uses the same experimental setup as for the overshoot scenario. Due to the strong overshoot of the parameterized skill, the initial guess for the selected task parameterization results in a bad start condition. Although the hybrid search is able to approach faster to higher rewards in the first half of the optimization process, the optimization by the standard CMA-ES is able to approach faster to the maximum of the reward function. As shown in Figure 4.8b, the optimization on the manifold of the parameterized skill results in a fast rising reward at the beginning of the optimization process followed by a period with slowly rising reward when it moves along the manifold to the defined optimum of the reward function.

Results

As shown in Figure 4.5-4.7, three situations can be identified in which the proposed hybrid CMA-ES algorithm is able to speed up optimization significantly. The mean rewards obtained for a given number of rollouts indicate a slightly faster convergence of HCMA-ES-v2, but the evaluation could not reveal a significant difference between HCMA-ES-v1 and HCMA-ES-v2 for those simple optimization tasks. Further, it was shown that in case of a faulty estimate of the parameterized skill in a region with low gradient information as well as a distortion of the manifold, the hybrid optimization scheme allows a faster convergence. Additionally, one situation in which the consistency of the parameterized skill can be enhanced by preference of solutions close to the previously established manifold was identified.

4.2.5 Evaluation on Robotic Scenarios

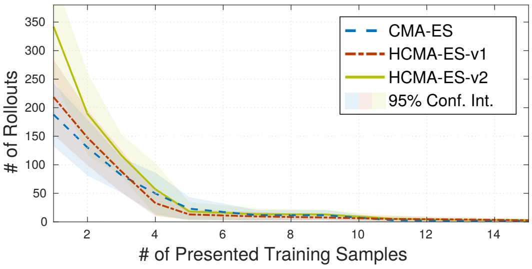


Figure 4.9: Results of the comparison of HCMA-ES to optimization in the parameter space of the policies for the point reaching scenario. It can be seen that the number of required rollouts for task fulfillment is not reduced significantly by one of the optimization methods.

Planar Arm Scenario The evaluation of the hybrid optimization scheme as proposed in Section 4.2.2 refers to the previously performed experiments as introduced in Chapter 3. The parameterized skill has to learn a skill for a point reaching task for a 10-DOF planar arm. The original CMA-ES optimization in policy space is compared to the hybrid optimization algorithms HCMA-ES-v1 and HCMA-ES-v2. To be able to compare the algorithms without the influence of different states of the memory, the stored memory states of the performed experiments in Section 3.4 have been used for evaluation. The following experiments replicate the same experimental conditions and replace the optimization algorithm by the proposed optimization in hybrid spaces. Figure 4.9 shows the results of the 10-DOF planar arm scenario. HCMA-ES-v2 requires slightly more rollouts for task completion than HCMA-ES-v2 and plain CMA-ES in case the memory has been trained with less than 4 samples. It can be assumed that it is caused by the overhead of updating the covariance matrix of the policy space based on rollouts in task space. To reduce the overhead of the hybrid search algorithms, the initialization of p_{E} and p_{F} plays a crucial role. It can be expected that a search in the policy space is more beneficial as long as the number of training samples for the parameterized skill is low. No substantial difference between the CMA-ES and the hybrid search can be seen in the case the parameterized skill consolidated more than 4 samples, The update policies of HCMA-ES-v1 and HCMA-ES-v2 do not lead to significantly different results.

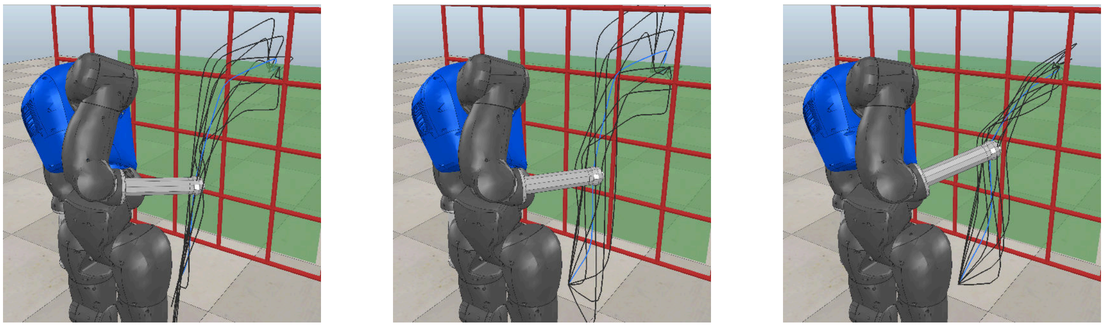


Figure 4.10: COMAN robot during execution of an estimated end effector trajectory (blue) of the parameterized skill PS(τ_i) for one fixed reaching target $\boldsymbol{\tau}_i$. Black trajectories visualize the variability in low-dimensional search space ±50% of the input range PS($\tau_i + \delta_{\pm 50\%}$). From left to right: different states of the memory are shown (3,5 and 10 training samples).

Point Reaching Scenario The results for the second scenario, Section 3.4.2, show that the proposed hybrid search is able to reduce the number of required rollouts for solving unseen tasks as expected. As before, memory states of the optimization in policy space are collected and reused for the comparison to the hybrid optimization approach to guarantee equal test conditions. The parameterized skill of the joint space experiments requires more training samples due to the lack of the inverse Jacobian controller that copes with distance maximization to the grid-shaped obstacle. The results are shown in Figure 4.12, (a-e) show results for experiments in end effector space in the same way as (f-j) show results for joint space. Both hybrid optimization methods show a tendency to exceed the rate of solvable tasks of the standard CMA-

ES method for the experiments in the end effector space Figure 4.12(b). The results of the joint space experiments the are not that clear Figure 4.12(g). The different update policies of HCMA-ES-v1 and HCMA-ES-v2 can be seen by a comparison of the development of the mixing factors p_{F} in Figure 4.12(c-e;h-j) for 1, 5 and 20 consolidated samples by the parameterized skill. In case the parameterized skill has been trained with all 20 samples , HCMA-ES-v1 switches to an optimization in the policy space at a later stage Figure 4.12(e+j), whereas HCMA-ES-v2 clearly prefers the policy space for optimization. Both algorithms are switching to a search in the policy space in case of a low number of training samples and in case the memory has seen a certain amount of training samples, HCMA-ES-v2 supports a faster switching from task to policy space search. The visualization of the variability in the low-dimensional parameter space is illustrated in Figure 4.10. The comparison of three different states of the parameterized skill is shown by plotting of estimated solutions for variations of the input centered at a fixed task parameterization. Those plots reveal different strategies of the robot like approaching the target point from top or from bottom, e.g. Figure 4.10(center).

Drumming Scenario The third scenario refers to the drumming task as presented in Section 3.4.3. In comparison to the previous experiments that evaluate the proposed hybrid optimization, evaluation was performed on a real robot scenario. Evaluation is limited to one exemplary state of the parameterized skill in which benefits of the hybrid optimization can be expected. The parameterized skill was trained with five samples that are gathered by kinesthetic teaching as before in Section 3.4.3. Therefore, the parameterized skill is in an intermediate state that allows to cover the structure of the task but does not result in a high success rate of about ~55% as shown in Figure 3.16. As in the previous experiment, the comparison of CMA-ES to both hybrid optimization methods HCMA-ES-v1 and HCMA-ES-v2 is performed for $N_{\mathrm{DOF}} = 8$-DOF. The reward function is based, as before, on the distance to the prototypes of the collected demonstrations and an additional punishment for trajectories that exceed the joint limits of the robot. The reward function R is defined by

$$R(\bar{f}) = \underbrace{log\left(1 + \max_{1\leq i\leq N_{tr}} \frac{\|\bar{f}_i^*\| - \bar{f} \circledast \bar{f}_i^*}{\|\bar{f}_i^*\|}\right)}_{\text{(a) Drumming Sound}} - \underbrace{\frac{\sum\limits_{\substack{0<t\leq T\\ 0<d\leq \bar{N}_{\mathrm{DOF}}}} \begin{cases} 0 & \text{if } q_{\min} < q_t^d < q_{\max}, \\ 1 & \text{otherwise.} \end{cases}}{T \cdot N_{\mathrm{DOF}}}}_{\text{(b) Joint Limits}} . \tag{4.14}$$

Maximization of the reward results in a minimization of the distance of generated drumming sounds to prototypes of the training set (details in Equation 3.11) as well as a minimization of joint angle trajectories that exceed the actuator limits $[q_{\min}, q_{\max}]$. The experiment is repeated ten times and for each experiment six unseen drum positions are selected for which the robot cannot play the drum successfully. For evaluation, the initial policy is estimated as $\boldsymbol{\theta}_{\mathrm{PS}} = \mathrm{PS}(\boldsymbol{\theta})$ and the number of required rollouts that are performed until the robot is able to generate a sound on the drum is collected. The results based on all 60 evaluation runs are shown in Figure 4.11. It can be seen that the hybrid optimization methods are able to solve unseen task instances with a significantly lower number of required rollouts for optimization.

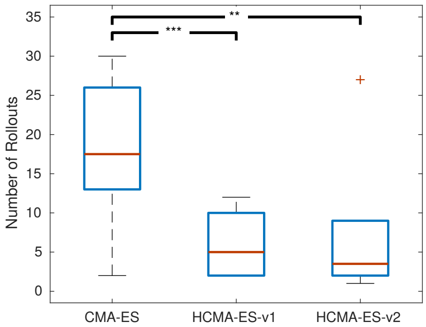
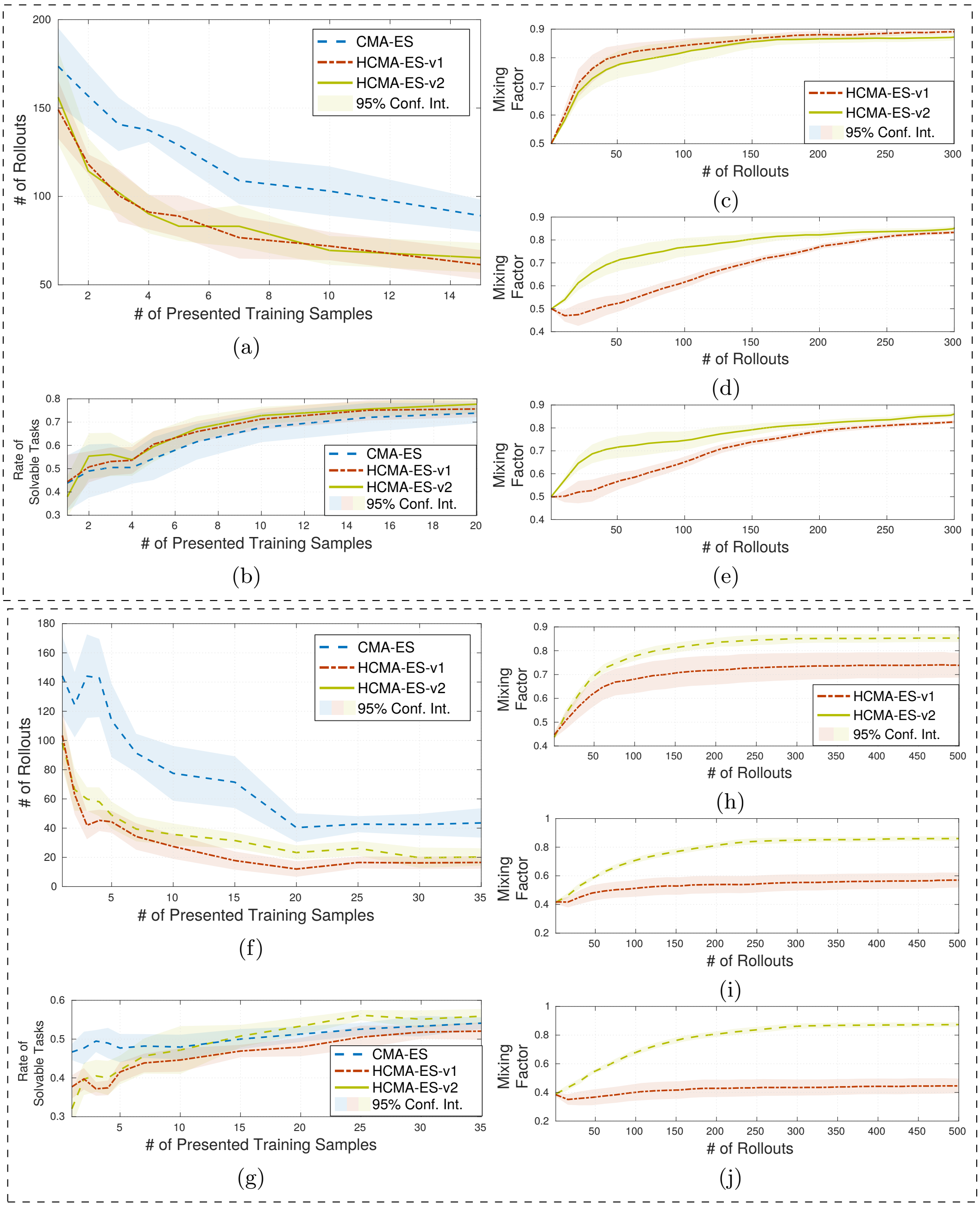


Figure 4.11: Evaluation of hybrid optimization methods on the Affetto drumming scenario. Results show the required number of rollouts for unsolved task instances.

4.2.6 Discussion

The experimental evaluation revealed three situations in which the proposed hybrid optimization was able to exceed the performance of an optimization in policy space, as discussed in Section 4.2.4. The benefits of the proposed algorithm as well as one case in which the proposed algorithm underlays a plain policy space search were shown for three designed test scenarios. A clear advantage of the proposed hybrid optimization could be identified, although the reduction ratio of the task space to the policy parameterization is only 2:1 for the idealized test cases. The scalability of the proposed method was evaluated in complex robot scenarios. It was not possible to show significant performance improvements of the hybrid search for the optimization of a 10-DOF robot scenario, while an optimization of a point reaching task of a humanoid robot showed the expected advantages of the approach. A possible cause for the low performance could be that the design of the 10-DOF reaching task, e.g. no obstacles, results in a simple reward function in the high-dimensional policy space. The optimizer in the full policy space is able to follow the gradient efficiently after an initial estimation of the covariance, e.g. direction, and a reduction of the search space is not beneficial. In such a situation, the algorithm is not able to exploit the benefits of the low-dimensional embedding of the parameterized skill and has to cope with overhead caused by the combination of both spaces. In case of the humanoid robot reaching task, the skill learning faces an optimization problem with a much higher complexity as it includes joint limits and obstacle constraints. Additionally, not all task instances are solvable due to kinematic constraints of the robot and CMA-ES cannot solve all tasks as it gets stuck in local minima. The demonstration of the benefits of the proposed combined optimization scheme for this complex scenario was successful. The approach was applied in different domains by an evaluation of control in joint and cartesian space and supports hypothesis **H4.1**. Besides the evaluation of reaching tasks, the scalability to real-world and online systems was demonstrated on a complex drumming task.

The proposed method is not limited to a trajectory encoding as DMP, an extension to rhythmic movements, for example, can be achieved by modification of the underlying DMP representation [Ijspeert et al., 2002]. Due to the modular design of the framework other policy representations, black-box-optimizer and learning algorithms can be integrated. One crucial benefit of the point attractor representation of the DMP is the linearity of its parameterization in relation to the task parameterization (e.g. target position). In comparison to e.g. vector field representations, instabilities can be avoided and the dimensionality of the policy parameterization is reduced. The system is designed to rely on the results of the optimization process, therefore it has no implicit capabilities of dealing with multiple objectives, like in e.g. [Pirotta et al., 2015; Parisi et al., 2017]. The pre-designed reward function has to reflect appropriate goals to fulfill the range of parameterized task instances. Policy estimation for multiple objectives can only be achieved by an encoding of the relevance of the objectives as task parameterization.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

Figure 4.12: Results of the comparison of HCMA-ES to optimization in the policy parameter space for the point reaching scenario. Experiments (a-e) show results in end effector space and (f-j) in joint space. The number of required rollouts for task fulfillment is significantly reduced by the proposed hybrid optimization method (a+f). The success rate of the optimization process (i.e. exceed a threshold on reward) stays the same compared to the optimization on the policy parameter space (b+g). In (c-e; h-j) the behavior of the mixing factor between the search spaces is shown for 1(c+h), 5(d+i) and 20(e+j) training samples.

4.3 Transfer Learning

Besides learning a completely new task, real-world situations often require the ability to adapt an already learned task to changing conditions without learning the acquired motion repertoire from scratch, as covered by **H4.2**. To investigate this issue, the humanoid robot Affetto (Section 5.2) learns to solve a drumming scenario (Figure 3.13) with varying positions of the drum as evaluated in Section 3.4.3. Then, the environment changes in a way such that the drum cannot be observed directly and the robot has to perceive the drum position through a mirror, located beside the workspace (Figure 4.15). Further potential changes in the scenario include the replacement of the original (possibly faulty) sensor by a newer/intact one, a changed position of the robot which would be otherwise static, or another modified point of view on the scenery. Relearning the complete task in the high-dimensional space of actions would be highly ineffective if instead the already acquired knowledge could be adapted and reused.

The field investigating such principles is called transfer learning [Pan and Yang, 2010; Salaken et al., 2017], in which the main goal is to reuse as much as possible of the previous knowledge for the new situation. Recently, a promising transfer learning approach has been proposed for classification in myoelectric prosthesis control under electrode shift [Paaßen et al., 2018]. This approach allows to transfer the classification model between two settings, without assuming a continuous drift, by optimizing a mapping of the input features directly for the target task.

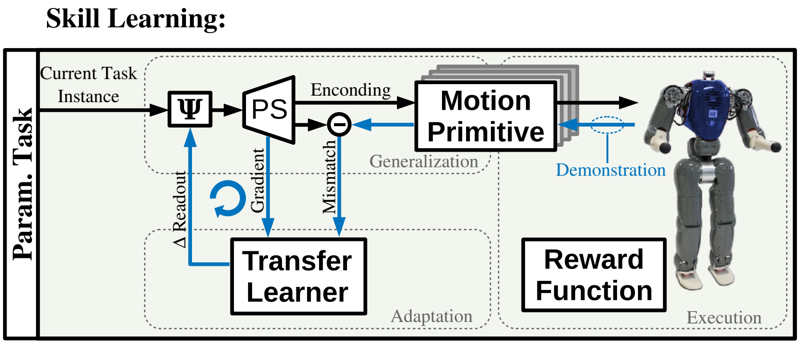


Figure 4.13: Illustration of the Transfer Learning approach. Based on human demonstrations, a transfer mapping ψ is updated according to the gradient of the parameterized skill.

In this section, the generalization of the transfer learning approach for a regression model is presented and applied for adaptation of a previously learned skill of a humanoid robot towards changing task conditions.

The remaining of this chapter is structured as follows. First, relevant related work on transfer learning, Section 4.3.1, and the proposed transfer learning algorithm are introduced. Section 4.3.2 illustrates the method for an artificial example while Section 4.3.3 describes the main experiment where the proposed transfer learning algorithm is employed to adapt towards a change in the environment for a drumming task.

Related Work The literature differentiates between different types of changing conditions [Pan and Yang, 2010]: Changes in the task and changes in the data domain. In this work, the latter case is considered where the task to be performed stays the same, while the data domain changes. In particular, the general assumption is that enough data are available in an old scenario, the so called *source domain*, but the goal is to solve the task in the new *target domain*, where only very few data are available. These types of problems are also referred to as *transductive transfer learning* [Pan and Yang, 2010] or as *domain adaptation* [Ben-David et al., 2006]. More formally, data instances from the source domain will be referred to as $\boldsymbol{\tau} \in \mathcal{T} = \mathbb{R}^E$ and to instances from the target domain as $\hat{\boldsymbol{\tau}} \in \hat{\mathcal{T}} = \mathbb{R}^{\hat{E}}$.

A popular set of methods in this area are related to the concept of importance sampling, one example being the kernel mean matching algorithm [Huang et al., 2007]. Those methods introduce weights for the data points in the source space and utilize them for learning a new supervised model to improve the performance in the target space. A central assumption is that the conditional distributions in both data spaces are the same: $p_{\hat{\mathcal{T}}}(\boldsymbol{\theta}|\boldsymbol{\tau}) = p_{\mathcal{T}}(\boldsymbol{\theta}|\boldsymbol{\tau})$ [Pan and Yang, 2010]. This strong assumption, however, does not hold in this scenario where the input space is changed strongly and thus the conditional distribution changes as well.

Another set of transfer learning methods aims to solve the transfer problem by finding a common latent space for the source and target domain [Pan and Yang, 2010; Blöbaum et al., 2015]. However, these methods assume the availability of only unlabeled data in the target space and, thus, do not make use of any supervised information if existing. Other work, such as Procrustes Analysis [Wang and Mahadevan, 2008], requires correspondence information between some samples from both domains which is unavailable for the drumming task.

Transfer learning has been applied in robotic settings, like reinforcement learning [Taylor and Stone, 2009]. For the purpose of multi-robot transfer learning [Helwa and Schoellig, 2017; Malekzadeh et al., 2014b], i.e. for learning a skill for a robot from another robot. A further application is inter-task learning, e.g. transfer knowledge of multiple acquired tasks to solve more complex new tasks [Fachantidis et al., 2012]. Those settings are, however, different from the presented ones because they consider only changes in the input but not in the output as learning is based on kinesthetic teaching to adapt for changing task configurations.

4.3.1 Transfer learning for nonlinear regression with the ELM

For formalizing transfer learning, this work follows the main idea from [Paaßen et al., 2018, 2016], which is to learn a mapping that transforms the novel target data in such a way, that the original model is applicable again. In contrast to [Paaßen et al., 2018, 2016], implementation aims at a regression model and is evaluated in a robotic scenario.

While in principle this technique is applicable to any supervised machine learning model with a differentiable cost function, demonstration is performed in this case on the regression model ELM. Given a training data set $\mathcal{D} = \{(\boldsymbol{\tau}^j, \boldsymbol{\theta}^j) | j = 1, \ldots, N_{\text{tr}}\}$ in the source domain, the ELM optimizes the cost

$$\sum_{j=1}^{N_{\text{tr}}} \sum_{i=1}^{F} \left(\boldsymbol{\theta}_i^j - \text{PS}_i(\boldsymbol{\tau}^j) \right)^2 \tag{4.15}$$

with respect to the parameters $\mathbf{W}^{out}$, where $\text{PS}_i(.)$ is defined in Equation 3.2. This results in a learned function $\text{PS}(\boldsymbol{\tau})$, applicable to instances from the source domain $\boldsymbol{\tau}$. A further discussion on ELMs and its learning methods is given in Section 2.2.2.

For the proposed transfer learning approach, the same cost function is utilized, but this time instances from the target domain are taken as input $\hat{\mathcal{D}} = \{(\hat{\boldsymbol{\tau}}^j, \boldsymbol{\theta}^j) | j = 1, \ldots, \hat{N}_{\text{tr}}\}$, with $\hat{N}_{\text{tr}} \ll N_{\text{tr}}$. Furthermore, the transfer mapping is defined as $\psi(\hat{\boldsymbol{\tau}})$ which is applied to the input $\hat{\boldsymbol{\tau}}$. Thereby, $\psi(.)$ realizes a mapping from the target to the source domain and learning its parameters comprises the main part of the transfer learning step. In many application, it is reasonable to assume a linear transformation of the form $\psi(\hat{\boldsymbol{\tau}}) = \boldsymbol{\Psi}\hat{\boldsymbol{\tau}} + \boldsymbol{b}$, where $\boldsymbol{\Psi} \in \mathbb{R}^{E \times \hat{E}}$ and $\boldsymbol{b} \in \mathbb{R}^E$. The transfer learning problem finally is

$$\min_{\boldsymbol{\Psi}, \boldsymbol{b}} \sum_{j=1}^{\hat{N}_{\text{tr}}} \sum_{i=1}^{F} \left(\boldsymbol{\theta}_i^j - \text{PS}_i(\psi(\hat{\boldsymbol{\tau}}^j)) \right)^2 + \gamma \|\tilde{\boldsymbol{\Psi}}\|^2. \tag{4.16}$$

Thereby, $\tilde{\boldsymbol{\Psi}}$ constitutes the matrix $\boldsymbol{\Psi}$ augmented by an additional column containing the values of $\boldsymbol{b}$ while λ is a weighting for the l2 regularization.

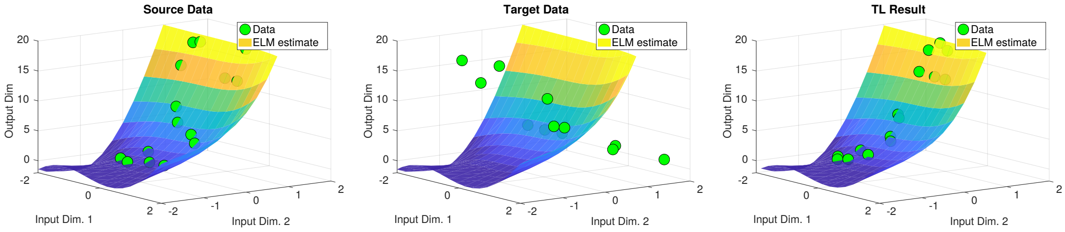
Then, finding a minimum of this problem with respect to the parameters of $\psi(.)$ constitutes the transfer learning step and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is employed for optimization.

4.3.2 Experiments I: A Toy Data Example

The Proposed transfer learning scheme is demonstrated for a toy data set first, before it is applied to a robotic setting in the next section. For evaluation, 20 data points are sampled from the function

$$\mathbb{R}^2 \mapsto \mathbb{R} : \boldsymbol{x} \mapsto (x_1 + 1)^3 + (2(x_2 + 1)^3)/10, \tag{4.17}$$

where 14 randomly selected points are utilized for training an ELM. The means squared error (MSE) is 0.00 on the training and 0.007 on the remaining testing data. The trained model together with the data is shown in Figure 4.14a. In order to simulate a systematic disturbance on the data, 20 new data points are sampled with an applied rotation of 180°. The resulting target data together with the original ELM



(a) (b) (c)

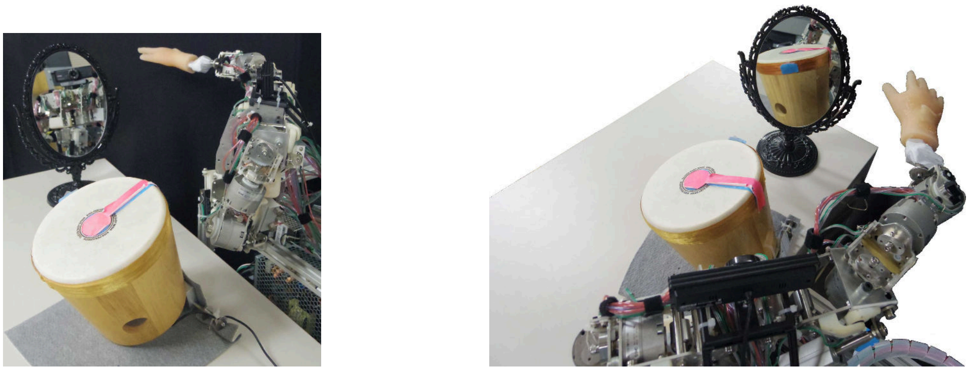
Figure 4.14: Illustration of the proposed transfer learning approach on toy data: the figures always show data (green circles) and the predictive function of the ELM (trained on the source data). (a) Source data; (b) Target data; (c) Target data after transfer learning.

is shown in Figure 4.14b. The prediction MSE is 585.772, due to the transformation of the new data.

For adaptation to the transformation of the data, five target data points are selected randomly and are used for training a transfer mapping with the proposed transfer learning algorithm. Using these transferred target data the algorithm can employ the original ELM to evaluate the quality of the transfer by calculating the MSE. Repeating this transfer step 100 times with different random training points yields the averaged MSE 0.001(±0.001) for the points used to train the transfer and 0.129(±0.381) for the other points (standard deviations in brackets). An example run is shown in Figure 4.14c. The median error of 0.035(±0.381) reveals outliers caused by local minima that disturb the gradient descent, therefore, the solution of $N_{\text{init}} = 10$ repetitions (random initializations) giving the lowest MSE is selected in the robotic setup.

4.3.3 Experiments II: Drumming Through Mirror on Humanoid Robot

This chapter aims at the evaluation of transfer learning for complex robot skills. The upper body of the humanoid robot Affetto has to play a drum positioned on a table in front of the robot, as shown in Figure 3.13. For the Transfer Learning condition of the experiments, the Affetto robot is not allowed to observe the drum directly and has to learn a new parameterized skill $\widehat{\text{PS}}$. As shown in Figure 4.15a, the robot is commanded to rotate its upper body into the direction of a mirror. As before, the marker position of the drum is extracted by blob detection. The rotation angle of the upper body is fixed, the task parameterization $\hat{\boldsymbol{\tau}} = (x_{\text{img}},\ y_{\text{img}})^\top \neq \boldsymbol{\tau}$ is given by the perceived location of the reflection of the marker in the mirror. Accordingly, there is a considerable difference in the mapping $\widehat{\text{PS}}(\hat{\boldsymbol{\tau}}) \neq \text{PS}(\hat{\boldsymbol{\tau}})$, so that relearning of $\widehat{\text{PS}}(\hat{\boldsymbol{\tau}})$ becomes necessary.



(a) (b)

Figure 4.15: Task parameterization of the modified perception in the drumming scenario.

Transfer Learning with Mirror

To solve this modified task, four learning schemes have been evaluated: i) the modification of the parameter space is ignored and the previously acquired parameterized skill PS is evaluated, as in Section 3.4.3; ii) relearning the task from scratch is performed in the same way as in Section 3.4.3; iii) the parameterized skill obtained in Section 3.4.3 is reused and training is continued with new human demonstration samples by incremental learning. Thereby ignoring the modification of the parameter space; iv) application of Transfer Learning as proposed in Section 4.3.1. Human demonstrations are utilized to estimate $\tilde{\boldsymbol{\Psi}}$ by application of Equation 4.16.

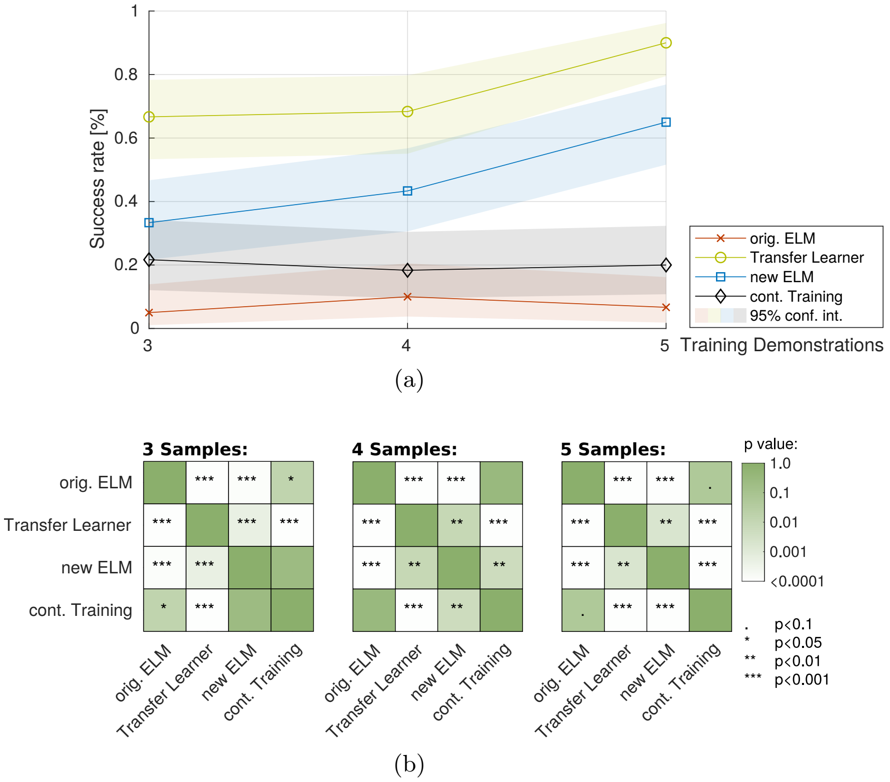
Let $\hat{\mathcal{D}} = \{(\hat{\boldsymbol{\tau}}^k, \boldsymbol{\theta}^k) | k = 1, \ldots, \hat{N}_{\text{tr}}\}$ be the new data set for transfer learning. Training is performed on $\hat{N}_{\text{tr}} = 6$ human demonstrations for drum positions distributed in the workspace of the robot. Each learner is incrementally trained with 3-5 randomly selected samples of $\hat{\mathcal{D}}$ and generalization performance is evaluated for 6 randomly selected unseen drum positions. The experiment is repeated ten times and the results of the evaluation can be seen in Figure 4.16.

Thereby, a baseline is given by the evaluation of the previously learned skill PS($\hat{\boldsymbol{\tau}}$) (i) resulting in a low performance due to the modifications of the task. Continued training of PS($\hat{\boldsymbol{\tau}}$) (iii) with new samples is also not able to adapt to the new task situation. A significantly better performance can be reached by transfer learning (iv) in comparison to relearning from scratch (ii).

4.3.4 Discussion

In this section, a novel transfer learning algorithm was presented, that aims at domain adaptation problems with a few labeled instances from the target domain and without correspondence information between the source and target space.

Evaluation of the method was performed on a toy data set for illustration and on a real world robot scenario in order to transfer complex motor skills. The approach significantly outperformed two baselines and a retrained model and supported hypothesis **H4.3**



(a)

3 Samples:

	orig. ELM	Transfer Learner	new ELM	cont. Training
orig. ELM		***	***	*
Transfer Learner	***		***	***
new ELM	***	***		
cont. Training	*	***		

4 Samples:

	orig. ELM	Transfer Learner	new ELM	cont. Training
orig. ELM		***	***	
Transfer Learner	***		**	***
new ELM	***	**		**
cont. Training		***	**	

5 Samples:

	orig. ELM	Transfer Learner	new ELM	cont. Training
orig. ELM		***	***	.
Transfer Learner	***		**	***
new ELM	***	**		***
cont. Training	.	***	***	

(b)

Figure 4.16: (a) Evaluation of the Transfer Learning approach against three test conditions: No update of the ELM for new situations, learning of a new ELM and continued training of the previous ELM. (b) Significance analysis of results for 3,4 and 5 presented training samples. Confidence interval is based on evaluation of 10 repetitions with 6 random unseen drum positions.

Chapter 5

Parameterized Skills for Compliant & Soft Robots

Chapter Overview *This chapter tackles the improvement of low-level control of highly compliant robotic systems by a combination of machine learning and classical control methods: First, an improved low-level control of pneumatic robots is presented that integrates an equilibrium model of the actuator. The inverse equilibrium model represents simplified properties of the dynamics of the robot, i.e. in case velocity and acceleration are zero. Second, an active compliant control mode that allows for kinesthetic teaching of highly compliant robots is proposed. Experimental evaluation was performed on a highly compliant, continuum soft robot and the humanoid robot platform Affetto. Further, the applicability of the proposed control mode for industrial light-weight robots is elaborated.*

This Chapter is Partially Based on:

- Queißer, J. F., K. Neumann, M. Rolf, R. F. Reinhart, and J. J. Steil
  2014. An active compliant control mode for interaction with a pneumatic soft robot. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pp. 573–579

- Rolf, M., K. Neumann, J. F. Queißer, F. Reinhart, A. Nordmann, and J. J. Steil
  2015. A multi-level control architecture for the bionic handling assistant. *Advanced Robotics*, 29(13: SI):847–859

- Balayn, A., J. F. Queißer, M. Wojtynek, and S. Wrede
  2016. Adaptive handling assistance for industrial lightweight robots in simulation. In *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, Pp. 1–8

- Malekzadeh, M. S., J. F. Queißer, and J. J. Steil 2017b. Imitation learning for a continuum trunk robot. In *Proceedings of the 25. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. ESANN 2017*, M. Verleysen, ed., Pp. 335–340. Ciaco

5.1 Compliant & Soft Robots

Bionic soft robots offer exciting perspectives for more flexible and safe physical interaction with the world and with humans. Unfortunately, their hardware design often prevents analytical modeling, which in turn is a prerequisite to apply classical control approaches. Further, also modeling by means of learning is hardly feasible due to many degrees of freedom, a high-dimensional state spaces and the softness properties, like e.g. mechanical elasticity, which causes limited repeatability and complex dynamics. Nevertheless, the realization of basic control modes is important to leverage the potential of soft robots for applications.

To tackle the challenges of control, this chapter presents a hybrid approach that combines classical and learning elements for an improved control and the implementation of an *interactive control mode*. The presented work evaluates control methods that superimpose a low-gain feedback control with a feed-forward control that is based on a learned simplified model of the inverse dynamics. To reduce the high-dimensional state space of the full inverse model of the robot, only equilibrium states of the robot are considered. It is demonstrated on the Bionic Handling Assistant (BHA), the humanoid robot child Affetto and an industrial light-weight arm, how a respective inverse equilibrium model can be learned and effectively exploited for quick and agile control. In a second step, the control scheme is extended to an active *compliant* control mode. It implements a kind of gravitation compensation to allow for kinesthetic teaching of the robot based on the implicit knowledge of gravitational and mechanical forces that are encoded in the learned model.

Compliant Robots for Human-Robot-Interaction As robotic systems with increasing complexity find their way into new application areas, the separation of human and robot work space is not feasible. Human-Robot Interaction (HRI) that aims for collaborative work or therapeutic use makes high demands on control architectures and the robot structure. One requirement is a safe operation, since the robot interacts with a human user, whereas classical stiff actuators have a high potential of injury. As Figure 5.1 shows, one way to lower the risk of injury is the reduction of the inertia of the moving parts of the robot. A second option to enhance safety of mobile manipulators is to lower the stiffness of the actuator.

In recent years, interactive robots that incorporate pneumatic and hydraulic actuators got more attention because of the inherent compliance of pneumatic actuation and the advantage to build light-weight actuators. This can be achieved by moving heavy parts into the torso of the robot in comparison to electrical actuators, which

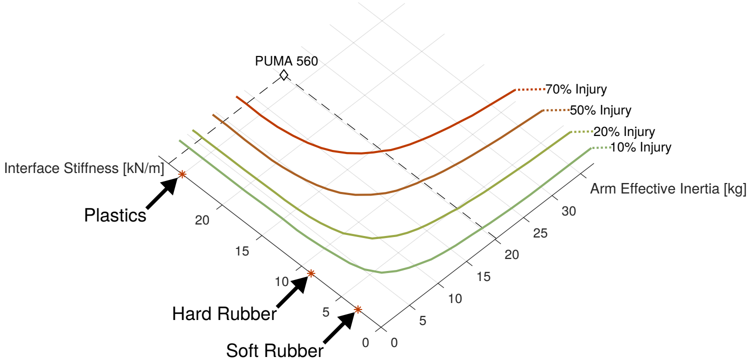


Figure 5.1: Relation of inertia and stiffness of a robot arm to the risk of head injuries on collision, data extracted from Zinn et al. [2004].

demand heavy and voluminous brushless motors as well as gears, close to the joint. One recent example is the actuator presented by Whitney et al. [2014]. They propose a light-weight antagonistic actuator based on diaphragm cylinders that is actuated by electronic motors located in the body.

Unfortunately the control quality of pneumatically actuated robots suffers from control delays, friction and complex dynamic properties. Modeling all these properties is difficult or sometimes even not possible at all and does not permit a reliable control. Moreover, additional environmental constraints, like the configuration of the robot or external forces affect model properties. So even by neglecting different load configurations of the actuator or contact forces during obstacle manipulation, the systems are faced with a high complexity of the control problem. Dealing with those challenges requires a dynamics model, parameterized by external configurations that may evolve over time by cause of e.g. changing material properties or task demands.

The work presented in this chapter aims at the experimental verification of the following hypotheses:

H5.1) The utilization of a learned inverse equilibrium model allows to improve the low-level control of highly compliant pneumatic actuators. (Section 5.3.1 & 5.3.2)

H5.2) A detection of a mismatch between the learned inverse equilibrium model and the real actuator signals supports the implementation of interaction modes on highly compliant and continuous robots. (Section 5.3.1 & 5.3.3)

In the following, the robot platforms will be introduced.

5.2 Robotic Platforms

The robotic experiments for evaluation of the inverse equilibrium model based control methods and the parameterized skills have been mainly performed on two platforms. In comparison to common robotic manipulators, both platforms are driven by pneumatic actuation and thus have complex nonlinear dynamical properties. This includes noise, control delays, changing material properties, parallel kinematics and high friction. Therefore, joint level control suffers from the lack of proper inverse models and results in a high tracking error. Those challenges makes the platforms an excellent study object for the proposed methods in this thesis. Successful task execution is not only based on a proper representation of joint trajectories, rather an additional representation of the properties of the dynamics is required to enhance the tracking performance and to allow for execution of precise movements.

Bionic Handling Assistant (BHA)

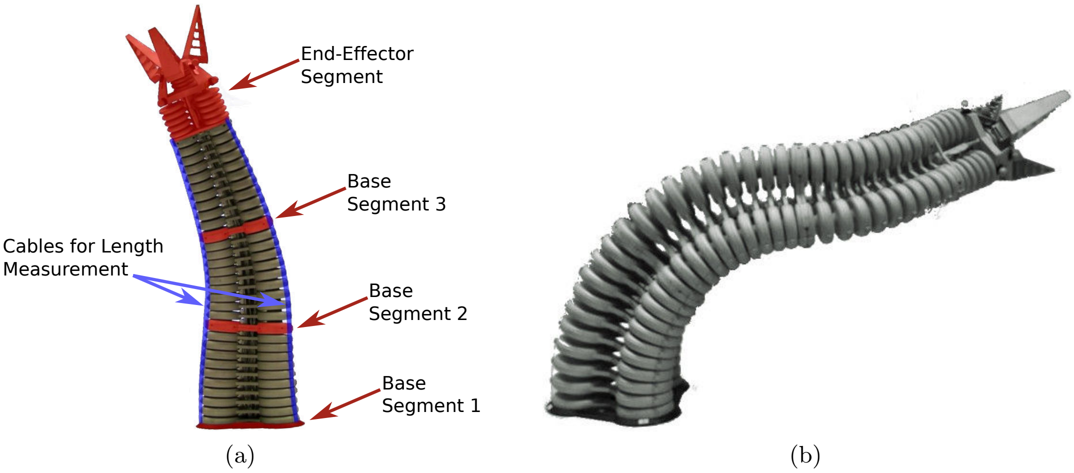


Figure 5.2: The Bionic Handling Assistant (BHA) robot. (a) Structural properties of the robot including length sensors. (b) Example posture with a deformation caused by a variation of the lengths of the pneumatic chambers.

The Bionic Handling Assistant (BHA) [Korane, 2010; Grzesiak et al., 2011] is a prominent, award-winning[1] robot platform designed by Festo as a robotic pendant to an elephant trunk. It displays typical challenges in soft robotics. The most significant challenge is induced by its novel actuation principle of co-activaion of three low-pressure pneumatic actuators in each segment that cause continuous deformations in shape. It has gathered strong interest because it belongs to a new class of continuum soft and light-weight robots based on low-priced and rapid 3D manufacturing with polyamide. It is pneumatically actuated and comprises several continuous parallel components operated at low pressures, which makes the BHA inherently safe for physical interaction with humans and provides a natural basis for future collaborative robotics tasks. This very properties distinguish it as a very mature representative from the field of continuum soft robots. The BHA robot is shown in Figure 5.2. Control of the BHA requires advanced algorithms that cope with the resulting redundancy, with non-stationarity due to the semi-fluid properties of the material,

[1] BHA won the prestigious German “Zukunftspreis” (future award) in 2010.

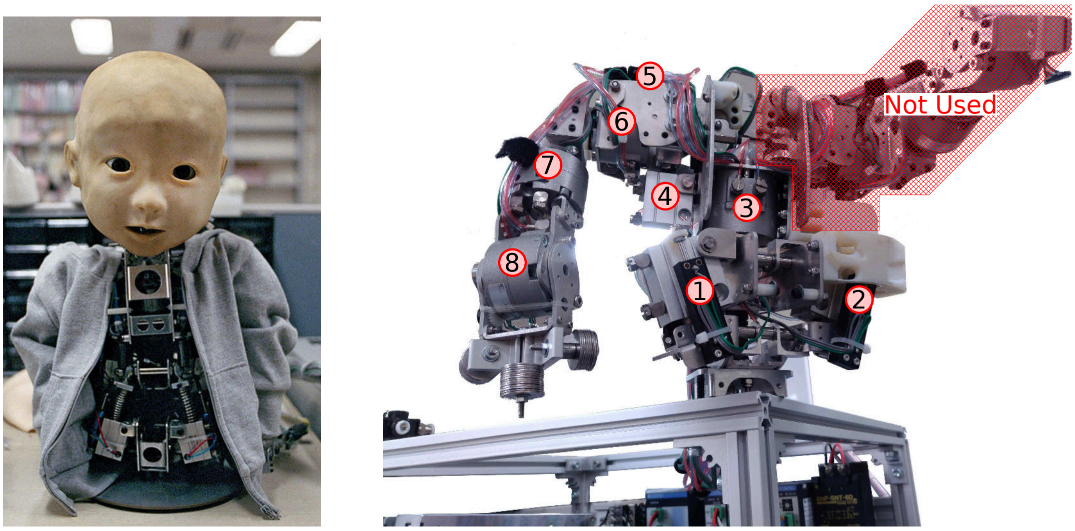
and the slow dynamics of the pneumatic actuation. The actuators operate with low-pressure pneumatics, which is not sufficient for a reliable control of the robot: air pressure only describes a force which results in a deformation of the structure of the robot. The structure of the BHA is separated into three segments (see Figure 5.2a). Each segment consists of three triangular arranged air chambers. Therefore the main flexibility of the BHA is based on nine air chambers that extend their length in relation to the pressures in those chambers. A fourth end effector segment is also available but was neglected for this work. An active depression of the pressure of the chambers is not possible, solely the tension of the extended body reforms the structure back to the home position. The robot has no fixed joint angles, each robot segment starts to bend in the case that the three chambers reach different lengths. Besides pressure sensors that are included in the air valves, the BHA is equipped with cable potentiometers that allow to measure the outer length of the air chambers that provide geometric information about the robot's shape. The segments of the BHA together with the attached cable length sensors are depicted in Figure 5.2a. In the following, the pneumatic actuators of each segment are considered separately and ignore the influence from the other segments. This approximation is reasonable because the inter-segment interaction is neglectable in comparison to the intra-segment interaction due to robot's morphology and its light weight.

Affetto Robot

The Affetto is a humanoid robot child driven by pneumatic actuators, as introduced by Ishihara et al. [2011]; Ishihara and Asada [2015]. It is developed in the frame of the **JST ERATO Asada Project**[2]. Affetto is modeled after a one- to two-year-old child for the purpose to study the early stages of human social development. But besides the child-like appearance, the Affetto robot unifies a wide range of motion, high compliance of the actuators and robustness. That makes the platform particularly interesting for experiments that target explorative learning, human-robot-interaction as well as the learning of low-level control signals. In particular, the robustness of the directly driven pneumatically actuated joints allows for unintended collisions with its own body structure or the environment during exploration without breaking the robot's structure. This is a crucial benefit in comparison to many of the high degrees-of-freedom humanoid robots used in research. Besides the upper body, recent work [Ishihara, 2016] introduces the full body joint mechanism that incorporates 26 pneumatic degrees-of-freedom. One particular challenge of the construction of a child-sized robot are space constraints. Pneumatic actuation allows for direct driven joints and the separation from the power source, i.e. compressor as well as valve unit, and the actuator itself. But as for the BHA robot, pneumatic actuation is challenging and results in complex dynamic properties. In particular for high-DOF robots, the lack of proper control hinders the applicability for pneumatic robots for complex applications. As an example, the 7-DOF pneumatic arm evaluated on

[2] Erato was the Greek Goddess of romantic poetry [https://www.jst.go.jp/erato/en/about/index.html]. ERATO is also a near-acronym for "Exploratory Research for Advanced Technology".

calligraphy writing introduced in [Hoshino, 2008], suffers from overshoots up to 10% of the joint range during the execution of dynamic motions. Further, robot systems that lack proper low-level control are for example a pneumatic robot based on artificial muscular-skeleton system as presented in [Ogawa et al., 2011] or a pneumatically actuated baby robot as proposed by Narioka et al. [2011].



(a)

(b)

Figure 5.3: (a) Pneumatically actuated humanoid child robot Affetto, as presented in [Ishihara et al., 2011; Ishihara and Asada, 2015]. (b) Joint configuration used for experimental evaluations of this thesis.

5.3 Inverse EQ-Models for Low-Level Control

The evaluation of hybrid control methods that incorporate an inverse equilibrium model of the plant will be presented in this section. Experiments have been performed on the continuous soft robot BHA, the pneumatically driven humanoid robot platform Affetto and a simulation of an industrial light-weight robot arm (UR5).

5.3.1 Bionic Handling Assistant (BHA)

Preliminary work of learning the equilibrium model of the BHA has been performed by Neumann et al. [2013]; Neumann [2014]. It introduced the integration of the equilibrium model into the low-level controller and presents an evaluation for step-responses. For this thesis, the learning of the equilibrium model was reproduced (**H5.1**) and a more exhaustive evaluation was performed. Additionally, a novel control mode that is based on the equilibrium model of the BHA will be proposed that provides a kinesthetic teaching mode and supports hypothesis **H5.2**.

Inverse Equilibrium Model for Length Control

A reliable and fast controller of the air chamber lengths is an indispensable prerequisite for the application of the BHA. The control can, in principle, be done with standard schemes like proportional integral derivative control (PID). The fundamental problem is that these approaches rely on quick and reliable feedback from the robot, while the BHA only provides very delayed and noisy feedback due to its pneumatic actuation and the visco-elastic mechanics. Consequently, the PID control can only be applied with low gains, which corresponds to slow movements.

An inverse dynamics model $\mathbf{f}_{\text{dyn}}^{-1}$ of the robot operating as feed-forward controller in addition to the low-gain feedback control could significantly decrease control delays. For the BHA, such an inverse model would map actuator lengths $\mathbf{l}$ and their derivatives $\dot{\mathbf{l}}$ and $\ddot{\mathbf{l}}$ to pressures $\mathbf{p}$ in the actuators.

$$\mathbf{p}(t) = \mathbf{f}_{\text{dyn}}^{-1}(\mathbf{l}(t), \dot{\mathbf{l}}(t), \ddot{\mathbf{l}}(t)) \tag{5.1}$$

However, the downside of the biologically inspired design of the BHA is that hardly any analytic models are available. Traditional approaches such as inverse dynamics based control becomes intractable. This fact qualifies learning as an essential tool for modeling, but collecting a data set that fully represents the inverse dynamics of the BHA is difficult.

Therefore, a simplified model is considered in this thesis, which is restricted to the mechanical equilibrium points $\mathbf{l}^*$ of the robot's dynamics. Equilibrium points are achieved by applying a constant pressure $\mathbf{p}^*$ until convergence of the lengths for a single segment. In such a state, neither lengths nor pressures of the pneumatic actuators change over time: $\dot{\mathbf{p}} = \dot{\mathbf{l}} = \ddot{\mathbf{l}} = 0$. The formulation of the inverse dynamics in Equation 5.1 thus simplifies to the following:

$$\mathbf{p}^* = \mathbf{f}_{\text{dyn}}^{-1}(\mathbf{l}^*, 0, 0) \Leftrightarrow \hat{\mathbf{p}}(\mathbf{l}^*) = \mathbf{p}^*, \tag{5.2}$$

where $\hat{\mathbf{p}}$ denotes the inverse equilibrium model that represents the direct relation between length $\mathbf{l}^*$ and pressures $\mathbf{p}^*$. The inverse equilibrium model provides a direct estimation of a reasonable control signal and can therefore serve as a feed-forward control signal that is applied immediately without waiting for delayed feedback.

A schematic illustration of this approach is shown in Figure 5.4. The image visualizes the BHA plant with its noisy and delayed feedback, the PID feedback controller, and the inverse equilibrium model. The BHA receives pressure commands, which are computed by a superposition of the low-gain PID controller and the feed-forward control signal from the inverse equilibrium model. The feed-forward controller computes pressures from desired length values by means of the inverse equilibrium model. PID control is based on the difference of the desired length values and the sensed length values. The PID controller thereby corrects the errors of the feed-forward controller in the feedback loop.

Learning the inverse equilibrium model from scratch is nevertheless difficult for several reasons: First, the underlying dynamics of the BHA result in a nonlinear

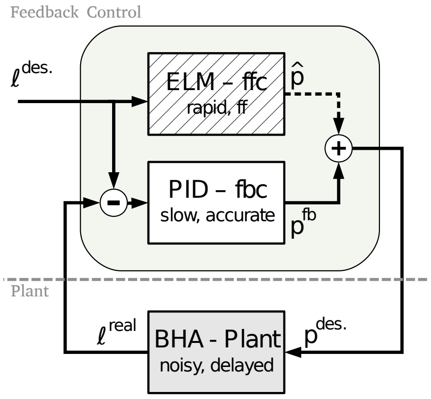


Figure 5.4: The control loop: combination of a learned inverse equilibrium model and a feedback controller. Leads to a fast estimation of the pressure configuration p^{des} for the chamber lengths L^{des}.

behavior which requires a model with appropriate complexity in order to capture the structure of the data to a sufficient degree. Second, data sampling is limited because the time until the physical deformations of the robot have reached a mechanical equilibrium can take up to 20 seconds for a single data point. Third, the resulting samples are very noisy due to physical hysteresis effects induced by the visco-elasticity of the robot's soft material. Finally, the material changes its properties due to the history of the manipulation. A predefined working area can change over time. Well-behaved extrapolation is thus a strong requirement on a learned inverse equilibrium model. Machine learning approaches which are trained on such data without additional efforts are prone to overfitting. To cope with these issues thus becomes an important requirement for the learner. The proposed approach of this thesis uses prior knowledge about the physical behavior of the BHA in order to derive a reasonable model of the length-to-pressures relation in a mechanical equilibrium in a data-driven manner [Neumann et al., 2013]. Note, that any other algorithm that can handle these requirements is potentially applicable.

Learning the BHA's Inverse Equilibrium Model with Prior Knowledge

For learning of an inverse equilibrium model, a machine learning approach is applied that is able to incorporate prior knowledge about physical constraints of the BHA in order to reduce over-fitting from few and noisy data and achieve well-behaved extrapolation. The following prior knowledge about the BHA is considered: (**i**) maximum and minimum pressure of the actuators are known in advance, and (**ii**) the ground-truth behavior per axis is strictly monotonous, because higher pressure in one actuator physically leads to an extension of this actuator.

Further, the observation that the entire mapping from length sensor values to chamber pressures can be separated into three, independent problems is used. This means that one inverse equilibrium model per segment (see Figure 5.2) is learned, which significantly reduces the demand for training data. This assumption neglects the gravity effects caused by a deflection of the remaining segments. However, these effects are rather small due to the robot's light weight and are corrected by the feedback controller.

The applied learning scheme is called Constrained Extreme Learning Machine (CELM) and was first introduced in [Neumann et al., 2013]. It is a feed-forward neural structure that comprises three layers of neurons. For the inverse equilibrium model of a single segment of the BHA, the CELM comprises $\mathbf{l} \in \mathbb{R}^{I=3}$ input, $\mathbf{h} \in \mathbb{R}^{N_\mathrm{H}}$ hidden, and $\hat{\mathbf{p}} \in \mathbb{R}^{O=3}$ output neurons. The input is connected to the hidden layer by the input matrix $W^{\mathbf{inp}} \in \mathbb{R}^{N_\mathrm{H} \times I}$. The read-out matrix is given by $W^{\mathbf{out}} \in \mathbb{R}^{O \times N_\mathrm{H}}$. For input $\mathbf{l}$, the output of neuron i is computed by

$$\hat{\mathbf{p}}_i(\mathbf{x}) = \sum_{j=1}^{N_\mathrm{H}} W_{ij}^{\mathbf{out}} f(\sum_{k=1}^{I} W_{jk}^{\mathbf{inp}} x_k + b_j), \tag{5.3}$$

where b_j is the bias for neuron j, and $\sigma(x) = (1 + e^{-x})^{-1}$ the logistic activation function. The components of the input matrix $W^{\mathbf{inp}}$ and the biases b_j are drawn from a random distribution and remain fixed after initialization.

Let $\mathcal{D} = (L, P) = (\mathbf{l}^k, \mathbf{p}^k)$ with $k = 1 \ldots N_\mathrm{tr}$ be the data set for training, where N_tr is the number of training samples. $L \in \mathbb{R}^{I \times N_\mathrm{tr}}$ is the collection of lengths, and $P \in \mathbb{R}^{O \times N_\mathrm{tr}}$ is the matrix of target pressures for all N_tr samples. Supervised learning is restricted to the read-out weights $W^{\mathbf{out}}$ and accomplished by solving a quadratic program which is subject to condition (**i**) and (**ii**) rephrased as linear constraints:

$$\|W^{\mathbf{out}} \cdot H(L) - P\|^2 + \alpha \cdot \|W^{\mathbf{out}}\|^2 \rightarrow \min \tag{5.4}$$

subject to:

$$(\mathbf{i}) \qquad \frac{\partial}{\partial l_i} \hat{p}_i(\mathbf{l}) > 0 : \forall \mathbf{l} \in \Omega$$

$$(\mathbf{ii}) \quad p_\mathrm{min}^s < \hat{p}_i(\mathbf{l}) < p_\mathrm{max}^s : \forall \mathbf{l} \in \Omega,$$

where $H(L) \in \mathbb{R}^{R \times N_\mathrm{tr}}$ is the matrix collecting the hidden-layer states. The growth of the read-out weights is controlled by the regularization parameter α. Ω is a predefined region in the model's input space, and $s, i = 1, 2, 3$ denote the segment and the output and input dimension.

The prior knowledge given by (**i**), (**ii**) defines inequalities on the learning parameters $W^{\mathbf{out}}$ at specific points $\mathbf{l}' \in \Omega$, which are sampled according to the approach in [Neumann et al., 2013]. Note that these inequalities are linear in $W^{\mathbf{out}}$. It was shown in [Neumann et al., 2013] that a well-chosen sampling of the points $\mathbf{l}'$ is sufficient for generalization of the point-wise constraints to the continuous region Ω.

Experimental Evaluation of Length Control

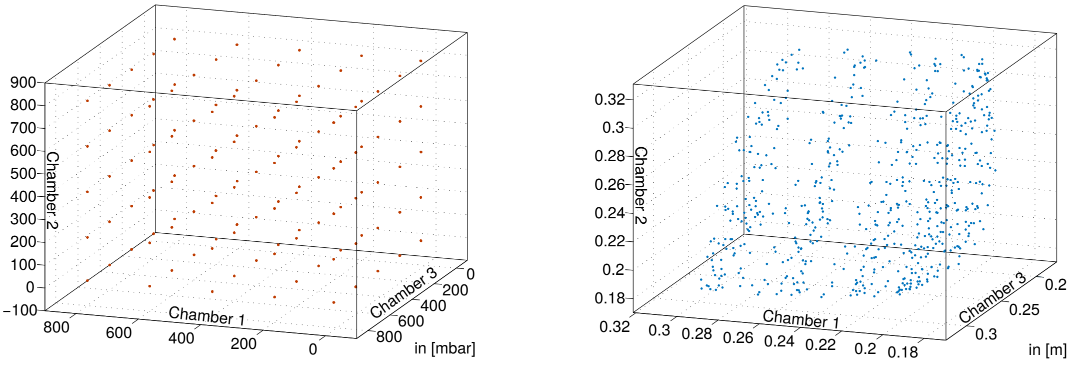
This section contains the results of a cross-validation test and the experimental evaluation of a linear model and the CELM model when applied in parallel to the PID feedback controller.

BHA Data Set For training of inverse equilibrium models, a data set is recorded. It captures the relation between the geometric length of the air chambers for each segment and the corresponding pressures in a mechanical equilibrium. Pressures are measured in milli-bar and the segment lengths in meters. For each segment, the pressure space is explored by applying pressures between minimum and maximum value in five equidistant steps. This results in a pressure grid comprising $5 \times 5 \times 5 = 125$ samples. For each pressure, the resulting combination of three lengths was recorded after a waiting phase of 20 seconds in order to reach the mechanical equilibrium. In order to deal with the inherent variation due to the visco-elastic material, this process is repeated five times with different traversal orderings, such that 625 samples per segment are available for learning. The minimum and maximum pressures, and the resulting length ranges are collected in Table 5.1.

Seg.	p_{max}	p_{min}	l_{max}	l_{min}	N	#Trials
1	800 mbar	0 mbar	0.32 m	0.17 m	625	5
2	1000 mbar	0 mbar	0.33 m	0.16 m	625	5
3	1200 mbar	0 mbar	0.32 m	0.16 m	625	5

Tab. 5.1: Properties of the BHA data set. Including pressure and length ranges of the segments of the actuator.

The grid for the applied pressures of segment 1 is illustrated in Figure 5.5a. The corresponding length values recorded on the robot are shown in Figure 5.5b. The data are nonlinear, with huge gaps in the middle part of the target data, for which generalization is critical.



(a)

(b)

Figure 5.5: Data set for chambers 1-3 of segment 1, each dimension represents one chamber. Pressure grid with five samples per dimension (a) and the corresponding length values (b). The nonlinear relation between lengths and pressures leads to gaps in the input space of the data.

Cross-Validation and Generalization The generalization performance of the learned models is evaluated on the BHA data set by cross-validation. Linear models (LM)

$$\hat{\mathbf{p}}(\mathbf{l}) = W^T\mathbf{l} + \mathbf{b} \tag{5.5}$$

trained by linear regression and the constrained ELM model (CELM) with additional use of prior knowledge are compared. An appropriate error measure for the learned inverse equilibrium models is the per-axis average-deviation from the measured ground truth value:

$$E = \frac{1}{N_{\text{te}}} \sum_{k=1}^{N_{\text{te}}} \frac{1}{D} \sum_{d=1}^{D} \|p_d^k - \hat{p}_d(\mathbf{l}^k)\|, \tag{5.6}$$

where N_{te} is the number of samples and $D = 3$ is the input and output dimensionality.

The results shown in Table 5.2 are obtained by cross-validation over the five trials measured by the error function in Equation 5.6. For each fold, four trials are used for training and one trial is used for testing the generalization ability of the models. Additionally, the errors are averaged over the five folds. The mean and standard deviation over the different cross-validation folds are presented in Table 5.2. The mapping ability of the LM is too poor to capture the structure encoded in the BHA data, the training (tr) and test (te) errors are large and indicate under-fitting. The CELM, in contrast, performs significantly better and is able to capture the nonlinearity of the mapping underlying the data.

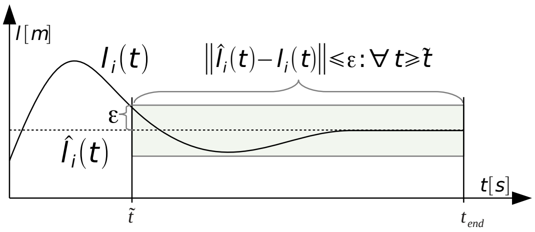
Segment	LM (tr / te) [mbar]	CELM
1	48.9±0.8 / 52.7±4.8	27.8±1.7 / 36.5±7.6
2	74.9±3.1 / 83.0±13.9	46.0±2.9 / 61.7±11.0
3	74.7±0.6 / 78.4±5.2	41.4+-1.6 / 54.6+-5.3

Tab. 5.2: Cross-validation errors of the BHA data set. Comparison between linear model (LM) and constrained ELM (CELM).

Experiment-In-The-Loop Experiments on the robot show the benefits of the learned inverse equilibrium model on the length control, as proposed by hypothesis **H5.1**. For a quantitative comparison, measure the time until convergence of the lengths to different target values is measured. Convergence is achieved after the desired lengths are reached with a certain accuracy ε as illustrated in Figure 5.6:

Given the target lengths $\hat{\mathbf{l}}$ and the measured lengths $\mathbf{l}$, convergence is reached at time $\tilde{t}$ if the error $\left\|\hat{l}_i(t) - l_i(t)\right\|$ for all actuators i is below the threshold ε for all time steps after $\tilde{t}$ until t_{end} is reached. For the experiments in this section $t_{end} = 10$ seconds was selected.

The convergence time was measured for each tested model on five random postures and repeated for ten times. Figure 5.7 shows the mean convergence time for



l[m]
$l_i(t)$
$\|\hat{l}_i(t)-l_i(t)\| \leq \varepsilon : \forall t \geq \tilde{t}$
ε
$\hat{l}_i(t)$
t[s]
$\tilde{t}$
t_{end}

Figure 5.6: Illustration of the convergence time measure. Convergence time is defined as the timespan $\tilde{t}$ whereupon the deviation of the measured lengths $l_i(t)$ of the chambers and the target lengths $\hat{l}_i(t)$ stays below the threshold ε until t_{end} is reached.

all trials. It is demonstrated that the length control without a feed-forward control signal, i.e. $\hat{\mathbf{p}} = \mathbf{0}$, requires a much longer convergence time than a feed-forward control with the linear model or the CELM. Furthermore, the figure shows that the CELM model benefits from its capabilities to model nonlinear data distributions in comparison to the linear model. It allows a more accurate prediction and thus a faster convergence in 68.25% of all sample postures for all tube sizes and in 82.22% for a tube size of $\epsilon = 0.35$cm.

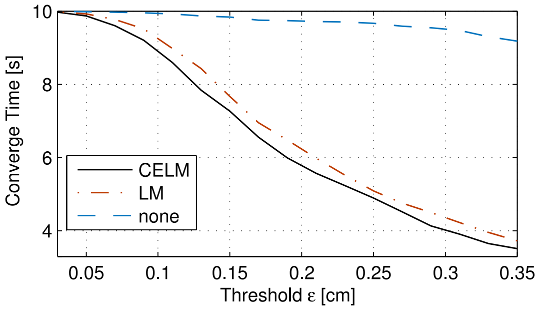
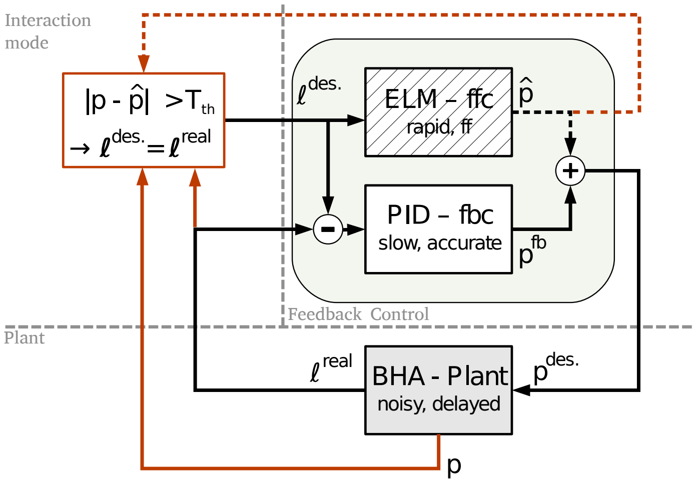


Figure 5.7: Convergence time of the length controller using different inverse equilibrium models: ELM with constraints (CELM), linear model (LM) and without inverse equilibrium model (none). Results are shown in relation to the tube size ϵ.

Equilibrium Model for an Active Compliant Control Mode

In comparison to the implementation of a *kinesthetic teaching mode* on stiff robots [Wang et al., 1998], a flexible robot structure allows the deformation of the actuator to a certain extent due to its softness. The detection of a deformation, e.g. caused by a human tutor, can be utilized to initiate a modification of the control variables such that the robot complies with the deformed configuration. The learned inverse equilibrium model of the robot can be used to detect deflections from the equilibrium by comparing the measured pressures of the chambers with the expected chamber pressures for the current lengths computed by the inverse equilibrium model. The control target lengths are then adopted such that the current configuration becomes the new equilibrium point of the robot. The resulting control mode allows for kinesthetic teaching of the robot as posed by hypothesis **H5.2**. This morphology-driven *external force detection* principle reduces the required computational effort and control complexity in comparison to classical approaches based on a full inverse dynamics model and accurate force sensing.

Figure 5.8 shows the interconnection of the different control modules to enable an active *compliant control mode* for the BHA. Essentially, the comparison between measured pressure in the pneumatic chambers $\mathbf{p}$ and the predicted pressures according to the learned inverse equilibrium model $\hat{\mathbf{p}}$ is added to the previous control scheme shown in Figure 5.4. Whenever the error $||\mathbf{p} - \hat{\mathbf{p}}||$ is above a predefined threshold T_{th}, the set-point $\boldsymbol{l}^{\text{des}}$ of the length control system is defined as the measured length sensor values $\boldsymbol{l}^{\text{real}}$. This leads to a redefinition of the set-point if the BHA is deflected from a mechanical equilibrium state. Such a deflection can be induced by a human interacting with the BHA and allows to deform the robot easily in an intuitive manner.



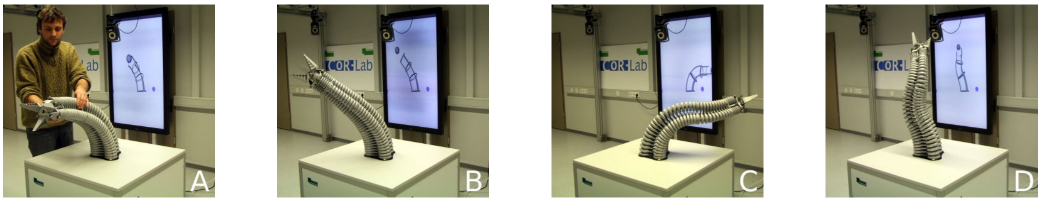
(a)

Figure 5.8: Active *compliant control mode* of the BHA achieved by application of a learned inverse equilibrium model of the pressure-to-length relation in a mechanical equilibrium.

A critical parameter for the functionality and the sensitivity of the *external force detection* is the threshold T_{th}. While smaller thresholds can result in drifts of the actuator due to inaccuracies of the inverse equilibrium model, larger thresholds limit the *interaction quality due to an improper detection of external forces* of the system.

Estimation of Threshold T_{th} In order to obtain a reasonable threshold T_{th}, a data set of 25 postures and the respective prediction error values was recorded.

Four postures of the data set and the corresponding errors are exemplarily shown in Figure 5.9. The average error of the learned inverse equilibrium model is 44.4 mbar, while the standard deviation is 8.9 mbar. The maximum and minimum error amounts to 58.7 mbar and 29.1 mbar, respectively. This motivates a threshold of $T_{\mathrm{th}} = 60$ mbar.



(a) (b) (c) (d)

Figure 5.9: Stable postures of the BHA after manual reconfiguration in active compliance mode. Manual reconfiguration of the BHA by a human tutor (A). Three exemplary postures from the test data set (B, C and D). Model Errors in the mechanical equilibrium: 50.6 mbar (B), 36.6 mbar (C), and 55.9 mbar (D).

Active Posture Control in Human-Robot Interaction Figure 5.10 shows a sequence of two manual reconfigurations of the BHA by a human tutor. The start configuration of the robot trunk is relaxed, the pneumatic actuators are deflated.

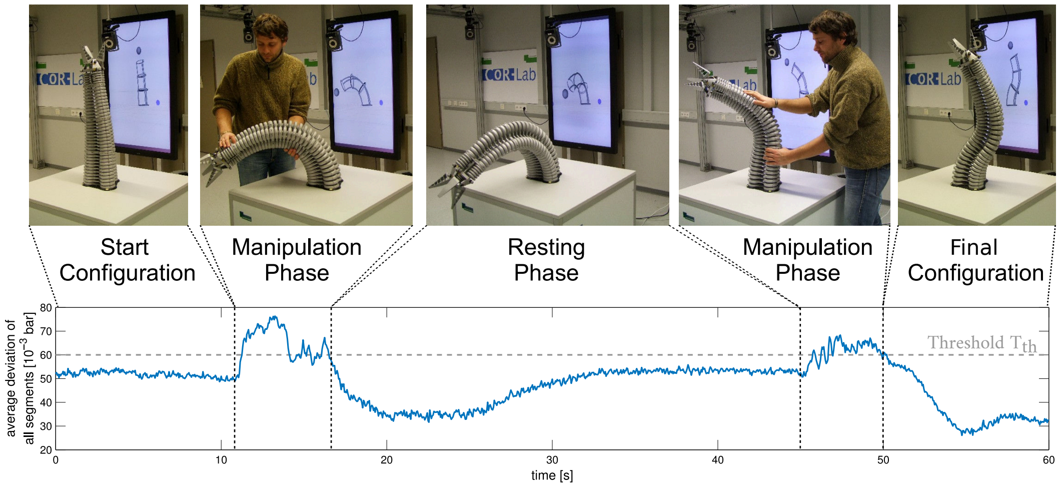


Figure 5.10: Active posture control in human-robot interaction. The graph on the lower part of the figure shows the prediction error $\|\mathbf{p} - \hat{\mathbf{p}}\|$ during human-robot interaction. The dashed line marks the selected threshold T_{th}. It is demonstrated that the prediction error exceeds T_{th} during the manipulation phase (adaptation) and falls below T_{th} during the resting phase (hold posture).

After roughly eleven seconds, the human operator starts to push the robot to the right side which deflects the robot's length and pressure state from the mechanical equilibrium point. This instantly induces an increasing prediction error $\|\mathbf{p} - \hat{\mathbf{p}}\|$ of the learned model $\hat{\mathbf{p}}$. When the error exceeds the threshold T_{th}, the set point of the desired length is reset to the current length sensor values. The length controller then adopts the pressures accordingly such that the current robot configuration becomes the new equilibrium point of the system. This tracking of the robot posture enables the user to easily change the posture of the robot trunk.

After a short time span, the robot again reached a mechanical equilibrium such that the error falls below the threshold (after approx. 16 seconds). During this time, the arm stays fixed until a second manipulation phase is started by the user (after 45 seconds). The manipulations lasts for five seconds and ends after the desired end posture is reached. The BHA stays stably in this position. This shows that the proposed control scheme is able to provide an useful *interactive* control mode without the need of complex internal models of the actuator. This human-robot interaction mode offers new fields of application for soft robots in research and practical applications.

Application Example The proposed active compliant control mode for the BHA robot was successfully applied on a parameterized pick-and-place scenario. The kinesthetic teaching mode allow the demonstration of movements with different target positions and rotations for an apple picking task. The complete action was divided into two primitives: 1) approaching an apple from a home position with respect to the position and orientation of the apple; 2) approaching to a basket for releasing the apple. The setup of the scenario is shown in Figure 5.12a. Generalization for new targets was implemented by Task Parameterized Gaussian Mixture Models (TP-GMM). The TP-GMM generalizes for unseen task instances by the combination of the represented demonstrations from the transformation of a set of frames. An illustration of a simplified setup is shown in Figure 5.11. For the representation of recorded demonstrations, Figure 5.11a, two reference frames are defined. For this example, the start position is defined as static reference frame and the goal position as a variable reference frame. For each reference frame's point-of-view, a GMM is estimated that encodes the demonstrations, as illustrated in Figure 5.11b. Temporal information that is required for trajectory generation can be encoded as an additional feature dimension of the GMM. An alternative is the representation as a decay term which yields a probabilistic representation of a DMP. Further details on GMM based representations are discussed in Section 2.2.2. For the selected example, it can be seen that the Gaussians of *Frame 1* have a higher precision (inverse of variance) close to the start point of the movement, whether the Gaussians of *Frame 2* have a high precision in front of the goal. Therefore, the model generalizes for movements from a start position to a goal, independently of the goal position as *Frame 2* encodes the trajectories from the coordinate frame that is assigned to the moving goal.

As introduced in [Malekzadeh et al.], the experiments for apple picking perform

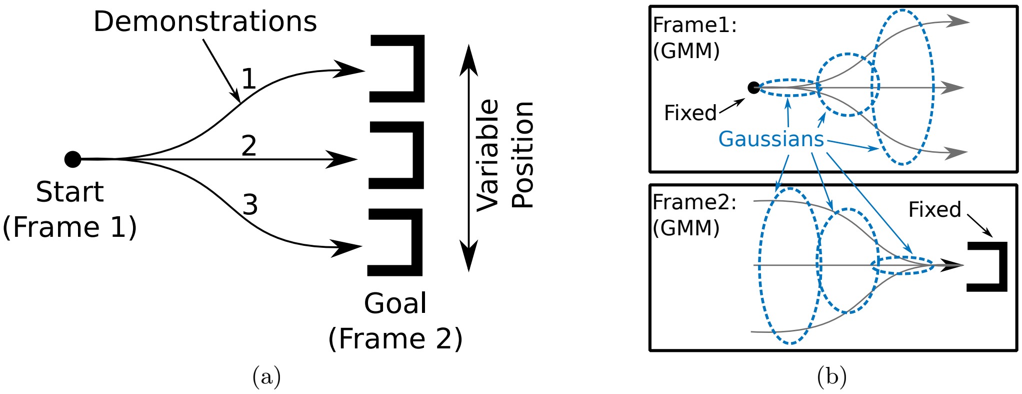
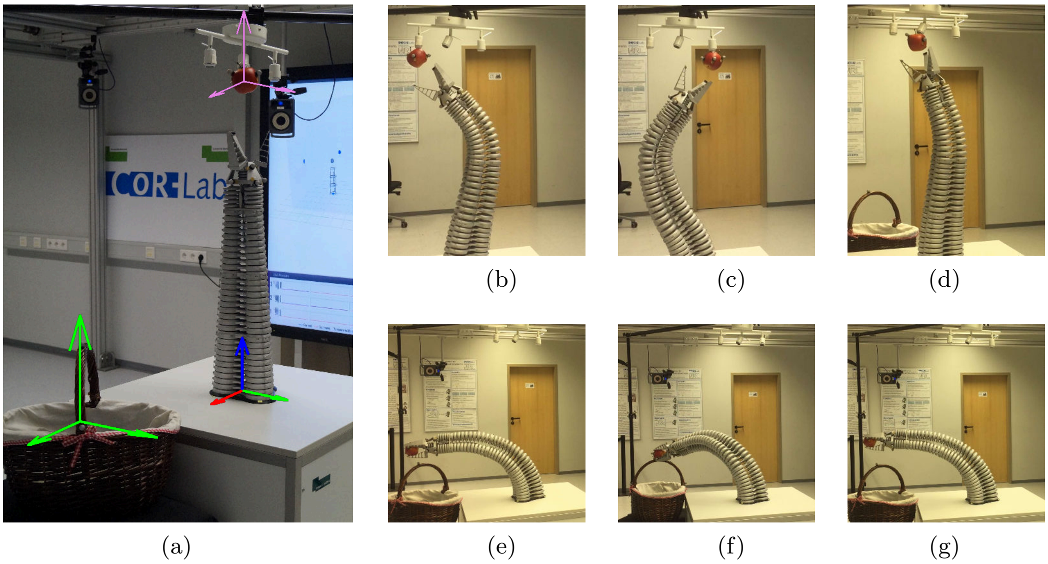


Figure 5.11: Illustration of the Task Parameterized Gaussian Mixture Model (TP-GMM). For training (a), multiple trajectories from demonstrations (1-3) are collected. Demonstrations cover the variability of the task parameterization, i.e. positions of the goal. Generalization is performed based on frames (b). For each frame one GMM is estimated that encodes the all demonstrations.

generalization based on three frames. In addition to the end effector position, the rotation is encoded as quaternions that allow an alignment of the gripper for the picking movement. More information on the implementation and the quaternion-based representation for TP-GMMs is presented in [Malekzadeh et al., 2017a; Malekzadeh et al.]. The results for generalized apple picking movements are shown in Figure 5.12. Generalized postures during the approaching of the apple for different rotations and positions are shown in Figure 5.12b-5.12d. The final posture for placing the apple into the basket can be seen in Figure 5.12e-5.12g.

5.3.2 Affetto

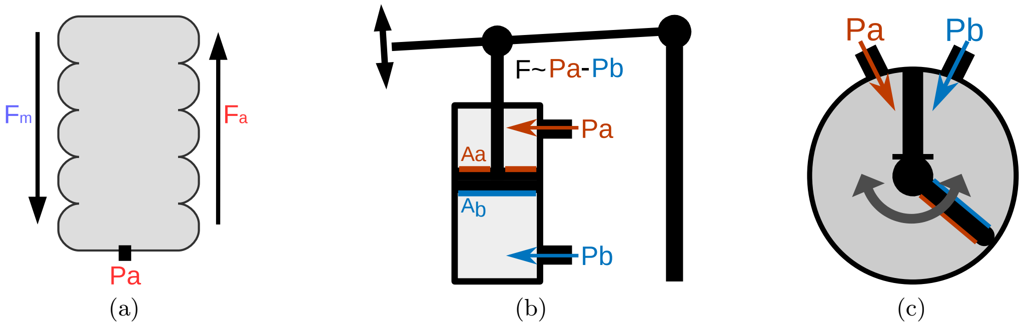
In comparison to the BHA, the Affetto robot incorporates classical air cylinders with an antagonistic actuation principle. The pressure difference in those chambers correlates to a force that is applied to the link. The basic working principles of all three pneumatic chamber types are shown in Figure 5.13. Each actuator of the BHA robot is based on one air chamber that extends by increasing the pressure Pa. The relation between the tension of the material that moves the actuator back to its initial shape and the force generated by the air chamber that elongates the actuator results in the output force of the actuator. Whereas the linear (Figure 5.13b) and rotary (Figure 5.13c) pneumatic actuators of the Affetto robot are driven by two antagonistic air chambers. The resulting force at the actuator is related to the difference of the pressures in the antagonistic chambers. The experiment was performed on the inner mechanics of the robot without assembled silicon skin parts that apply an additional force during bending of the joints. Therefore, the equilibrium model of the Affetto robot mainly represents the compensation for gravitational forces to remain at stable



(a) (b) (c) (d) (e) (f) (g)

Figure 5.12: (b-d) Different poses in apple reaching: the learned model was used successfully to reach apples with different positions and orientations. (e-g) Different positions of basket in apple picking: the model for the second part of the experiment was examined in different situations.

postures. Note, it is expected that the proposed approach scales to changes of the configuration of the robot, e.g. in case of an attached skin, learning of the robot's inverse equilibrium model can be done in the same way as described in this chapter. The control of the robot is achieved by proportional valves, a commanded voltage



(a) (b) (c)

Figure 5.13: Illustration of pneumatic actuator concepts.

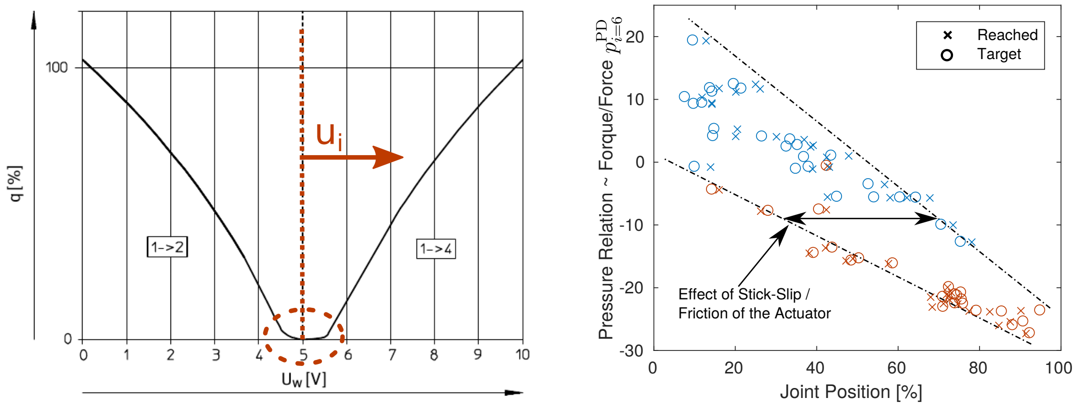
controls the piston position in the valve which results in a variable the opening diameter. Depending on the pressure difference, the opening of the valve leads to an air-flow and thus a variation of the pressure in the chamber. Each chamber is connected to a two-way valve that allows to open a channel for each chamber to the compressor to increase the pressure or open to the environment to decrease the pressure in that specific chamber. The response characteristic of the pneumatic valves is shown in Figure 5.14. The feedback controller for the presented experiments is based on the PIDF controller as introduced in [Todorov et al., 2010]. The controller extends the classical PID controller scheme for the antagonistic actuation principle of the pneumatic actuators. The controller signals that are sent to the valve are estimated by

$$u_i^+ = k_F \left[\underbrace{\left(k_P \left(q_i^*(t) - q_i(t)\right) + k_D \left(\dot{q}_i^*(t) - \dot{q}_i\right) + k_i \int_0^t \left(q_i^*(s) - q_i(s)\right) ds \right)}_{\text{PID Controller } p_i^{\text{PID}}} - \underbrace{\left(A_{\overrightarrow{ab}}\, pa(t) - pb(t) \right)}_{\substack{\text{Pressure Difference,} \\ \text{Represnets Force } p_i^{\text{PD}}}} \right] . \quad (5.7)$$

The factor $A_{\overrightarrow{ab}} = {}^{A_a}\!/_{A_b}$ compensates for the relation of the active areas A_a and A_b of the antagonistic chambers. It ensures a $p_i^{\text{PD}} = 0$, in case both chambers generate an equal force. In case of a piston, Figure 5.13b, the rod reduces the active area of one chamber resulting in $A_{\overrightarrow{ab}} \neq 1$. The rotary actuator, Figure 5.13c, incorporates equally sized active areas and thus results in $A_{\overrightarrow{ab}} = 1$. A PID controller p_i^{PID} operates in the domain of target pressure differences of the antagonistic chambers based on the joint error q_i. The commanded p_i^{PID} is compared to the measured pressure difference p_i^{PD} and results in the control signal u_i^+ that adjusts the opening of the valves. The relation between valve opening and control signal is shown in Figure 5.14a.

The controller can be interpreted as a nested PID controller that is wrapped by an outer proportional controller with gain k_F that controls the pressure difference of the antagonistic chambers. This allows the inner PID controller to operate in the force domain as the pressure difference correlates with the force of the actuator. [Todorov et al., 2010] show that this control method results in a lower tracking error as well as a lower time delay for trajectory tracking tasks compared to classical PID control. They evaluate the PIDF controller on 2-DOF of a high-DOF pneumatically actuated humanoid robot that has a comparable actuation principle as the Affetto robot.

The evaluation of the PIDF controller on the hardware of the Affetto robot reveals static control offsets. The effect is exemplary shown for joint #4 in Figure 5.15. For each joint, different static offsets can be observed. It can be assumed that manufacturing tolerances of the valves are the main cause, since the offset is not affected by the commanded trajectory nor the joint position. In particular, the



(a) (b)

Figure 5.14: Characteristics of the proportional valve (a) used for the control of the Affetto robot. The voltage at the valve is given by $U_A = 5.0 + u_i^+$ respectively $U_B = 5.0 + u_i^-$ for each antagonistic actuator. Evaluation of the effects of friction on the control of the pneumatic actuators (b). Color indicates the direction ($\dot{\mathbf{q}}_{i=6} \geq 0$) for approaching the target position for controller signals $p^{\mathrm{PD}}_{i=6}$.

transition from fully closed to a slightly open valve has a highly nonlinear relationship and is not symmetric, as highlighted by the red dashed circle in Figure 5.14. To compensate for these disturbances, the implemented controller for the Affetto platform introduces a dynamic offset compensation with a slow adaptation rate α: $p_i^{offset}(t+1) = \alpha p_i^{offset}(t) + (1-\alpha)(p_i^{\mathrm{PID}} - p_i^{\mathrm{PD}})$, with $\alpha = 10^{-3}$. The resulting control signal, used for the control of the valves, is given by

$$u_i^+ = k_F(p_i^{\mathrm{PID}} + p_i^{offset} - p_i^{\mathrm{PD}}) \tag{5.8}$$

and vise versa $u_i^- = -u_i^+$, for the control of the valve of the antagonistic chamber.

Inverse Equilibrium Model for improved Low-Level Control

As for the BHA, PID control can only be applied with low gains due to the pneumatic actuation. Long tubes connecting the control unit with the actuators have to be assembled inside the body of the child-sized robot. Additional sensory noise requires low pass filtering and causes further control delays. Due to the similarities in actuation principle, the experimental section will compare the performance of the basic controller against an extended controller that incorporates a learned feed-forward signal based on the inverse equilibrium model of the robot, as successfully applied for the BHA. Instead of the representation of the chamber pressures in relation to the posture of the robot, the inverse equilibrium model of the Affetto robot represents the relation of the chamber pressures, represented in p_i^{PD}, due to the antagonistic actuation principle. Therefore, the data set for training is given by

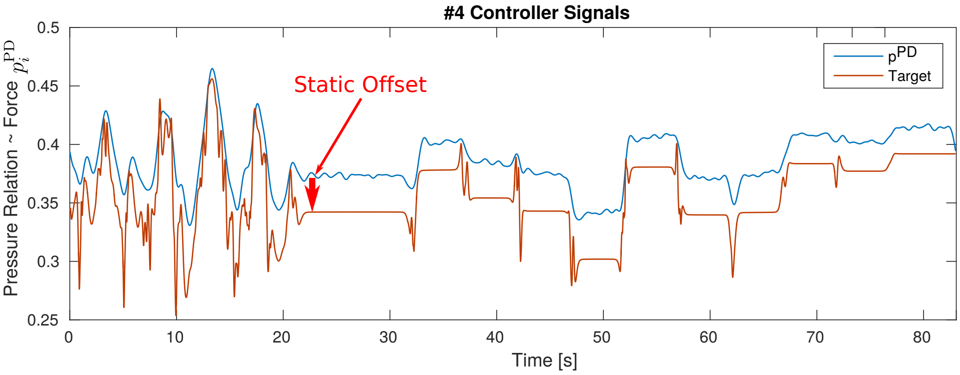


Figure 5.15: Visualization of the evaluation of the joint controller for joint #4. A static offset between the desired and reached pressure difference signal can be seen. Each joint controller has a different offset. Each joint shows an independent offset.

$\mathcal{D} = (L, P) = (\mathbf{q}^k, \mathbf{p}^{PD,k})$ for $k = 1 \dots N_{\text{tr}}$ training samples. In the same way as before, weights W^{out} and bias $\mathbf{b}$ of an ELM

$$\hat{p}_i^{\text{PD}}(\mathbf{p}) = \sum_{j=1}^{N_{\text{H}}} W_{ij}^{\mathbf{out}} f(\sum_{k=1}^{I} W_{jk}^{\mathbf{inp}} q_k + b_j) \quad , \tag{5.9}$$

are estimated with respect to the minimization of the error of a recorded training set. In the same way as for Equation 5.6, ridge regression was applied. As in the previous experiment, input dimensionality is $I = 3$ and number of output neurons $\hat{\mathbf{p}}_i^{\text{PD}} \in \mathbb{R}^{O=3}$ are are defined. The number of hidden neurons $\boldsymbol{h} \in \mathbb{R}^{N_{\text{H}}}$ and the regularization γ of the readout weights are estimated by a grid search. No additional constraints for the optimization are used, because a monotonic relationship of p_i^{PD} and q_i cannot be guaranteed. The resulting controller that incorporates the inverse equilibrium model is defined as

$$u_i^+ = k_F(p_i^{\text{PID}} + p_i^{offset} + \hat{p}_i^{\text{PD}}(\mathbf{q}) - p_i^{\text{PD}}). \tag{5.10}$$

In case of the BHA, the force to overcome the friction of the actuator is much lower than the forces occurring in equilibrium points. The Affetto robot suffers from the friction in the actuators as it has a high relevance in relation to the gravity and disturbs the training of the inverse equilibrium model. As an example, a recorded evaluation data set reveals the huge influence of the actuator's friction, as shown in Figure 5.14b for the measured control signal p_i^{PD} for equilibrium states $\mathbf{q}_i$. Depending on the actuator's movement direction to approach target joint states, distinct levels of p_i^{PD} can be observed. Positive movement directions, i.e. $\dot{\mathbf{q}}_i \geq 0$, are marked red, otherwise blue color is used. In Figure 5.16, a more detailed evaluation of the relation between joint positions of stable states and recorded control signals p_i^{PD} is shown. In the lower part of the figure, the commanded joint positions and the real positions are shown. In the upper half, the temporal aligned control signals that represent the force, p_i^{PD} and p_i^{PID}, are plotted. The joint is commanded to follow a square wave from 40% to 60% of the actuator's range. The blue coloring highlights stable states at 40% and the red coloring highlights stable joint states at 60% of the joint range. It can be seen that the range of p_i^{PD} values in the stable states of the two joint positions overlap. Additionally, it can be seen how the pressure relation of the pneumatic chambers changes (e.g. at time 30sec.) to compensate for the friction of the actuator until a overshoot of the controller can be compensated by the integral part of the PID controller of p_i^{PID}.

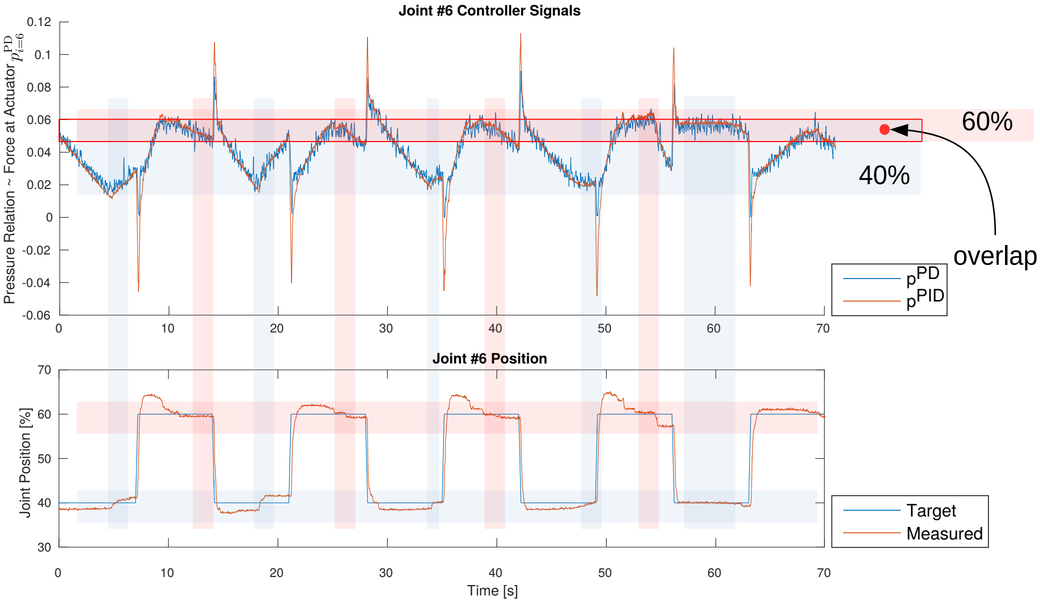


Figure 5.16: Controller signals for an executed square wave trajectory. Red and blue areas highlight the min and max values of the joint position. An overlap of the controller signals for both states is highlighted by a red square.

To improve the response time of the slow integral error compensation, a further extension of the low-level controller is proposed. It is assumed that in equilibrium states, i.e. tracking error is zero as well as $\dot{\mathbf{q}}(t) = 0$ and $\ddot{\mathbf{q}}(t) = 0$, the inverse equilibrium model $\hat{\mathbf{p}}^{\mathrm{PD}}(\boldsymbol{q})$ compensates for the integral component of the controller p_i^{PID}. This assumption allows to perform a reset of the error integration of the PID controller to zero on direction changes of the actuator. This thesis will refer to the controller without the reset of the integral component of the PID controller and no additional equilibrium model as *PIDF*, to the controller that embeds the inverse equilibrium model as *PIDF_EQ*, and to the controller that embeds the equilibrium model and the reset of the integral component as *PIDF_EQ_I_RESET*.

Affetto Data Set For evaluation of inverse equilibrium model, a data set of the right shoulder (joint #4-6, see Figure 5.3b) is recorded. It captures the relation between posture $\boldsymbol{q}_i$ of the robot and the corresponding values of $\boldsymbol{p}_i^{\mathrm{PD}}$ in a mechanical equilibrium, similar to the experiments of Section 5.3.1. The training data set consists of $N_{tr} = 500$ samples of random joint angle configurations in the range 5-95% of the joint range. To ensure the recording of equilibrium states of the robot, recording takes place after a duration of 2 seconds with no movement, i.e. $||\dot{q}|| < 1$. The recording process estimates the mean value of ten successive recordings.

Cross-Validation and Generalization The evaluation of the generalization capabilities of the of the learned models of the Affetto data set is performed by cross-validation. Optimal parameters for

- regularization $\gamma \in \{1e^2, 1e^1, 1e^0, 1e^{-1}, 1e^{-2}, 1e^{-3}, 1e^{-5}, 1e^{-6}, 1e^{-10}, 1e^{-12}, 1e^{-14}, 1e^{-16}\}$
- hidden layer size $N_\text{H} \in \{10, 25, 75, 100, 125, 150, 175, 200\}$

are estimated by grid search, as shown in Appendix A.1. The result of the cross validation of the inverse equilibrium model on the training set (tr) and test set (te), for $N_\text{H} = 125$ hidden neurons and a regularization of $\gamma = 1$, can be seen in Table 5.3. As for the BHA experiment, the results are compared to a linear model. The results indicate that the data set size is sufficient for the learning problem, as the difference between training and test error is below 10^{-2}. Moreover, the model benefits from a nonlinear approximation since the ELM is able to achieve a slightly lower error rate.

Joint #	LM (tr / te) [*bar*]	ELM
1	13.52±0.25 / 13.52±0.94	10.54±0.48 / 10.54±1.9
2	5.49±0.10 / 5.49±0.38	3.67±0.1 / 3.67±0.52
3	16.51±0.16 / 16.51±0.64	11.57±0.34 / 11.57±1.34

Tab. 5.3: Cross-validation errors of the Affetto data set. Comparison between linear model (LM) and ELM (ELM).

Trajectory Tracking Experiments The following experiments aim at an experimental verification of hypothesis **H5.1**. For each evaluation of the three controller variants, an optimization of the controller gains is performed. The optimization is performed in a semi-automatic way by initial hand tuning and a successive automatic optimization procedure as in [Todorov et al., 2010]. For the automatic optimization of the parameters k_P, k_I, k_D and k_F of the feedback controller, a simultaneous independent optimization for each joint is performed. The reward for optimization of the controller parameterization is given by

$$R_i(\mathbf{q}_i^*, \mathbf{q}_i) = \min_{0 \leq t_{shift} \leq t_{span}-1} \left[\frac{1}{T - t_{shift}} \sum_{t=t_{shift}}^{T} \left(\mathbf{q}_i^*(t_{shift}) - \mathbf{q}_i(t - t_{shift}) \right)^2 \right]. \tag{5.11}$$

The reward estimates the minimum tracking error shifted within a time-span of 200 milliseconds, $t_{span} = 60Hz \cdot 0.2\text{sec}$. For each rollout of the optimization by CMA-ES, the joint controller executes a predefined joint trajectory $\mathbf{q}^*$ that is composed out of sine-wave and step responses. To protect the robot in case of an unstable controller configuration, the joint trajectory is limited to 5-95% of the joint range. In case the robot's joints exceed this limit and reach a joint configuration of 0-2% or 98-100%, the joint controller is deactivated for the current rollout and a low reward is given. For safety reasons, the optimization procedure was under human observation and in case of unstable controller configurations human intervention was possible. Although no human intervention was necessary during the optimization process. The result of the automatic optimization process can be seen in Figure 5.17. Starting with the initial hand tuned parameters as initialization, the automatic optimization process is successful in finding significantly better controller parameters for trajectory tracking. In all cases of the test set, the controller that utilizes the equilibrium model with an additional reset of the integral part (PIDF_EQ_I_RESET) resulted in the lowest tracking error. The PIDF controllers with and without utilization of the equilibrium model (PIDF & PIDF_EQ) reach a similar performance, except for the fourth joint. For the final evaluation of the tracking error of the actuator, a trajectory tracking task was evaluated, as shown in Figure 5.18. The figure shows the target and reached trajectories for each joint. The sample rate is $f_{\text{sample}} = 60Hz$ with a duration of $T = 85$ seconds for the execution of the evaluation trajectory. For better visibility, parts of the recorded data are shown with 4x magnification in the black rounded rectangles. A similar performance of the PIDF controller with and without utilizing the inverse equilibrium model can be seen. For the PIDF controller that utilizes the inverse equilibrium model and performs an additional reset of the integral part of the controller (PIDF_EQ_I_RESET) less overshoots in case of the step responses as well as a faster response following the sine wave can be observed. A detailed qualitative evaluation of the three controllers is presented in Figure 5.19. The tracking error $E_{tracking}(\mathbf{q}^*, \mathbf{q}^{real}, t_{shift})$ for each joint in relation to the delay of the comparison t_{shift} is shown. The evaluation was performed with

$$E_{tracking}(\mathbf{q}^*, \mathbf{q}, t_{shift}) = \frac{1}{T - t_{shift}} \sum_{t=t_{shift}}^{T} \left(\mathbf{q}^*(t) - \mathbf{q}(t - t_{shift})\right)^2 . \tag{5.12}$$

The results reveal that the PIDF_EQ_I_RESET controller (extended by the equilibrium model and an additional reset of the integral component) is able to reach the lowest tracking error for all joints. Additionally, the proposed controller is able to improve the time delay for the point of the lowest tracking error for two joints #5 and joint #6. The PIDF controller with the additional inverse equilibrium model (PIDF_I_RESET) seems to suffer from high sensory noise (caused by mechanical construction) of joint #4, for the remaining joints, the controller is not able to improve tracking performance and reaches a comparable performance as the PIDF controller.

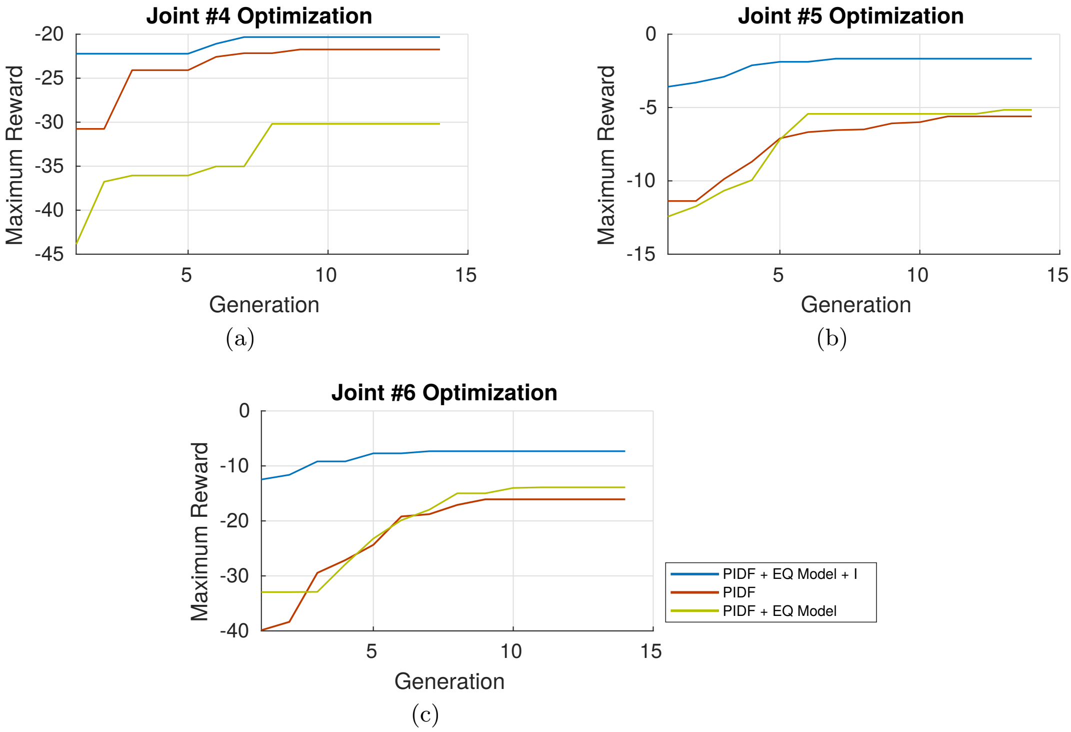
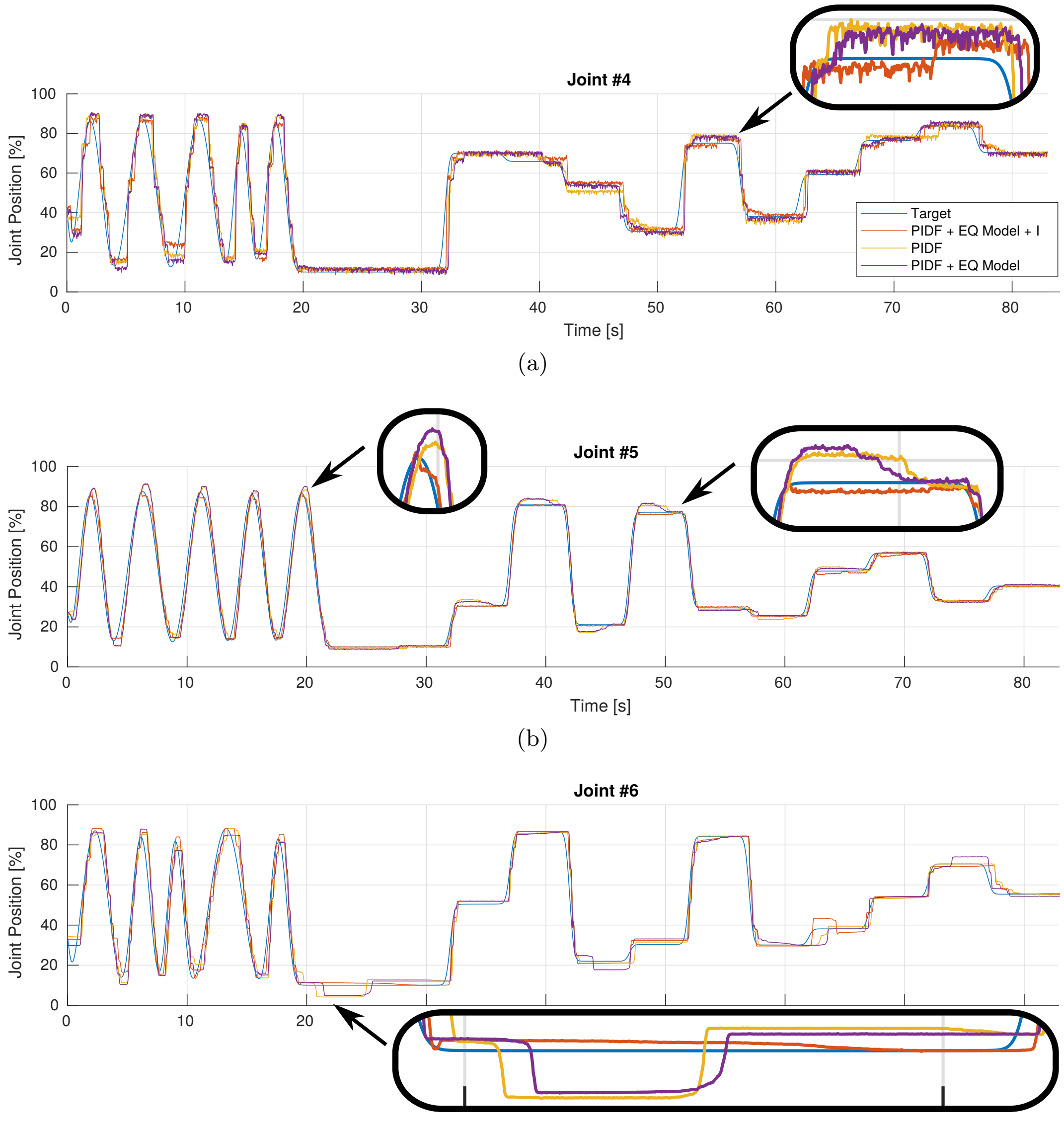


Figure 5.17: Results of the semi-automatic optimization process of the PID controller gains. For each joint one independent optimization was performed in parallel. Optimization starts with a hand tuned parameterization and reaches a significantly higher reward by automatic optimization.

Scalability to high-DOF Configurations The experiments that are presented in the following utilize up to 8-DOF of the Affetto robot platform. They implement the PIDF_I_RESET controller as introduced in the preceding section as it reaches the lowest tracking errors. For each configuration of the robot that was used throughout this thesis, an equilibrium model was recorded and parameter optimization of the controller was performed. The results for the most complex configuration of 8-DOF with an attached rubber hand (see Section 3.4.3) is shown in the following Table 5.4.

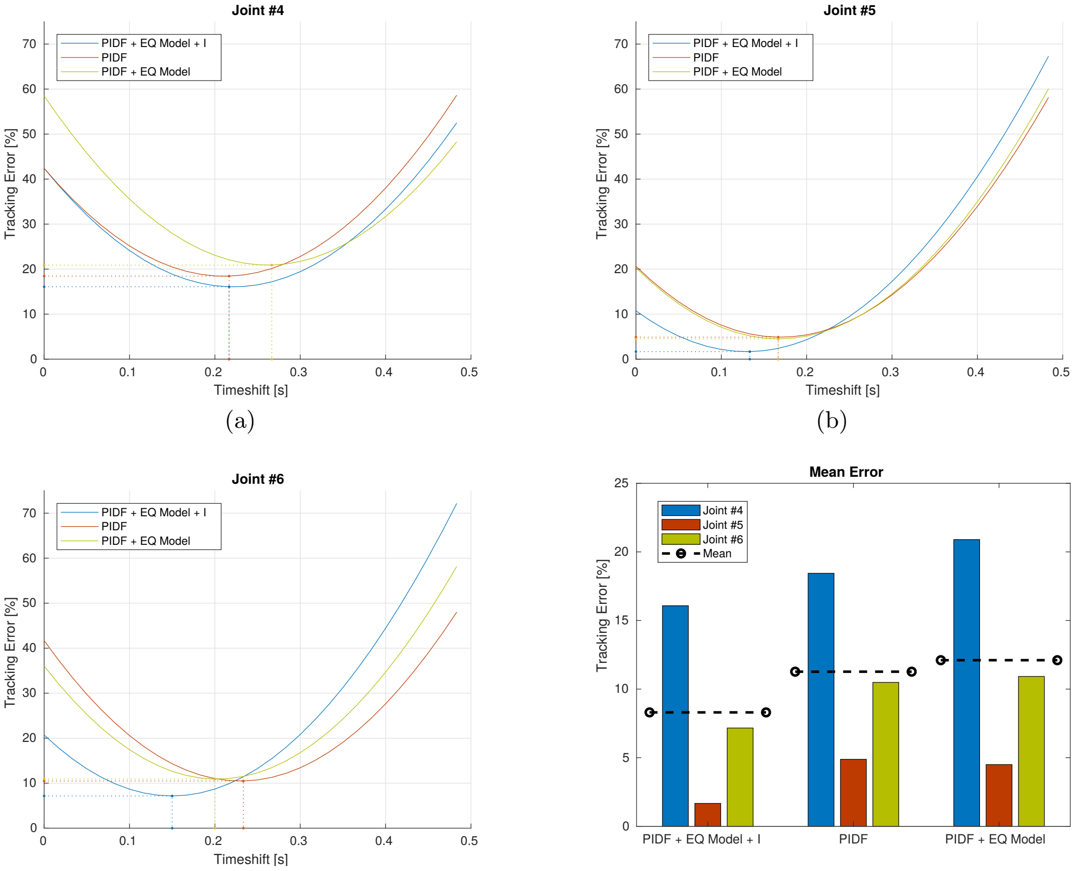


(a)

(b)

(c)

Figure 5.18: Evaluation of three proposed controllers based on extensions of the the PIDF controller as introduced by Todorov et al. [2010]. Experiment evaluates three joint of the right arm (a-c). Trajectories are limited to 5-95% of the joint range. Rounded squares highlight details with 4x magnification.



(a) (b)

(c) (d)

Figure 5.19: Evaluation of the tracking error in relation to the time delay of the proposed controller. Results show the evaluation of three joints of the right arm of the Affetto robot (a-c). In (d), the comparison of lowest reachable tracking error for each controller is shown. Units are mean errors in [%] of the joint ranges.

Controller:	#1	#2	#3	#4	#5	#6	#7	#8
PIDF	2.14 ±.05	3.54 ±.06	3.38 ±.08	2.71 ±.05	1.84 ±.04	4.38 ±.08	4.51 ±.10	5.24 ±.10
PIDF + EQ	1.62 ±.04	1.91 ±.04	4.00 ±.09	3.29 ±.07	1.68 ±.04	3.93 ±.08	4.56 ±.10	4.81 ±0.10
PIDF + EQ + I	1.44 ±.04	1.27 ±.03	2.73 ±.06	2.95 ±.06	1.51 ±.04	2.93 ±.06	3.23 ±.08	3.55 ±.07

Tab. 5.4: Comparison of tracking performance with the PIDF controller, and the PIDF controller extended by inverse equilibrium model (PIDF_EQ) and additional reset of the integral component (PIDF_EQ_I_RESET). Units are mean errors in [%] of the joint ranges.

5.3.3 Compliant and Lightweight Industrial Robots

This section investigates the applicability of the previously proposed interactive control mode, for industrial light-weight robots. Typically, industrial robots are equipped with high quality dynamics models which makes learning approaches or hybrid methods for low-level control obsolete. But in case of compliant robots in an industrial context, quick adaptations to new working environments are necessary. Due to the light weight and the high compliance of the robot, equipment that is attached to the actuators, like stiff tubes, cables or protective skins, have a huge impact on the dynamics of the robot system and yield motivation to transfer the methods developed for soft robots to industrial platforms. In the following, the application requirements for robots in industrial contexts will be discussed in detail, prior to the introduction of a system that incorporates an equilibrium based control mode.

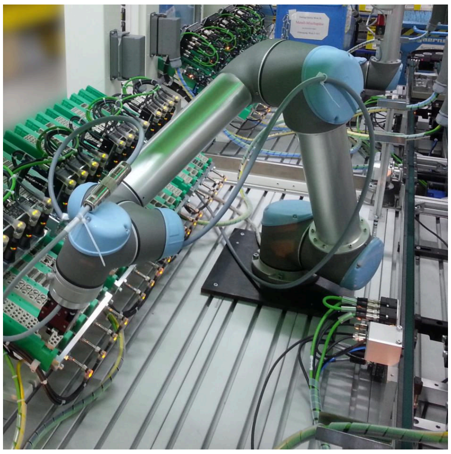


Figure 5.20: UR5 light-weight robot in industrial context. Constrained working environment and additional sensors are shown.

Light-weight Robots in Industrial Contexts

A flexible production that targets small lot sizes requires flexible usage of industrial light-weight robots. Light-weight robots [Popić and Miloradović, 2015] are used in production systems to close automation gaps where a full automation is not profitable, reasonable or possible. Further, they are supposed to support humans in manufacturing tasks. Therefore, robots and their control systems offer modes to enable intuitive interaction with the human worker. Common modes for interaction are, for example, gravity compensation and joint impedance modes. These modes are able to react to external forces and compensate the effect of gravity on the robot. For instance, the human worker operates in gravity compensation to perform pick-and-place tasks of heavy objects with assistance of the robot. In that case, gravity compensation balances the weight of the robot and the load with a counter force to make it moveable with less effort. However, gravity compensation requires exact knowledge about the weight of the robot and the work loads. Otherwise, motions of the robot become unpredictable and dangerous for the human worker and the payload. To deal with this problem, the weight of the tool as well as the manipulated object have to be declared to the robot system in advance. This additional effort is a drawback for flexible robot systems that have to adapt quickly to frequently changing tasks in modern manufacturing environments. Although methods for model estimation and parameter search for models of rigid body dynamics exist, e.g. [Vuong and Jr., 2009; Goto et al., 2003; Ding et al., 2015], they cannot be applied in all cases. For example, a high flexibility in the workplace design of the robot leads to an individual configuration setup like stiff cables, soft grippers or protective shields that are attached to the actuators. Without an appropriate model of those structures, uncertainties in the dynamics of the robot can occur. A proper application of light-weight robots is not feasible in this situation.

Control Architecture

The control architecture for the implemented compliant control mode is depicted in Figure 5.21, it is designed according to the previous setup of the BHA, presented in Section 5.3.1. The equilibrium model (1) estimates the expected torque $\hat{\boldsymbol{\tau}}$ for the current configuration $\mathbf{q}^*$ of the robot. The interaction module (3) estimates a desired posture update $\mathbf{q}$ based on the error (2) of the predicted $\hat{\boldsymbol{\tau}}$ and real torques $\boldsymbol{\tau}^*$ of the robot. In case an error threshold is exceeded (4) the current target joint angles (5) are updated or kept unchanged. This ensures that no drifts of the robot actuator occur that are caused by noise or inaccuracies of the learned the equilibrium model. The integrated PID position controller (6) updates the real torques $\mathbf{q}$ that are sent to the simulator (7) and the robot model.

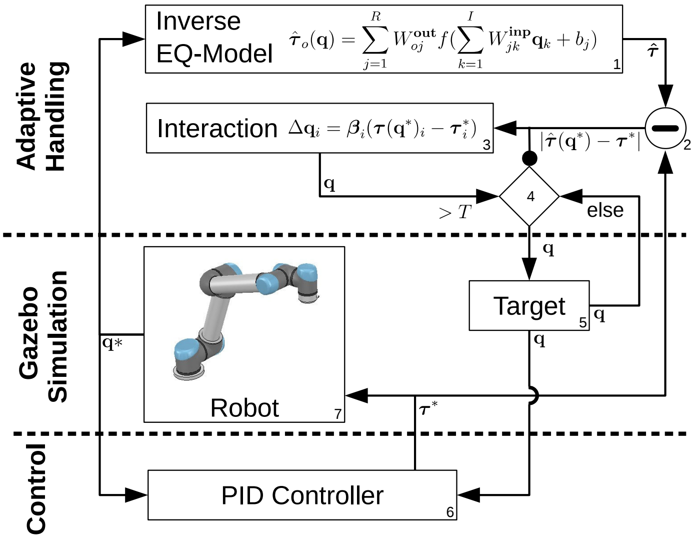

Adaptive Handling
Inverse EQ-Model $\hat{\boldsymbol{\tau}}_o(\mathbf{q}) = \sum_{j=1}^{R} W_{oj}^{\mathbf{out}} f(\sum_{k=1}^{I} W_{jk}^{\mathbf{inp}} \mathbf{q}_k + b_j)$ 1
$\hat{\boldsymbol{\tau}}$
Interaction $\Delta \mathbf{q}_i = \boldsymbol{\beta}_i(\boldsymbol{\tau}(\mathbf{q}^*)_i - \boldsymbol{\tau}_i^*)$ 3
$|\hat{\boldsymbol{\tau}}(\mathbf{q}^*) - \boldsymbol{\tau}^*|$
2
q
$> T$
4
else
Gazebo Simulation
q
Target 5
q
q
q*
Robot 7
$\boldsymbol{\tau}^*$
Control
PID Controller 6

Figure 5.21: Proposed control architecture for an inverse equilibrium model based adaptive control mode on light-weight robots. Inverse equilibrium model (1), estimation of prediction error (2), posture update based on prediction error (3), threshold based activation of posture update (4), desired joint angles (5), position controller (6) and robot simulator (7).

Software Architecture

This section describes the software architecture that visualizes, simulates and controls the UR5 robot. For the experiments with the adaptive control mode, the robot simulator Gazebo[3] is used to simulate the real robot behavior. In general, the software architecture is implemented within the robotic framework OROCOS (Open Robot Control Software Project)[4] and its Real-Time capable Toolkit (RTT)[5]. RTT allows an inter-component communication with output and input ports that exchange data between the components. In this case the components mostly provide and receive a six dimensional joint vector corresponding to the six rotational joints of the UR5 robot.

Moreover, the software architecture for the compliant control mode of the UR5 represents a component structure that distributes functionalities of the framework to each component. Components which are taken into account for handling an unknown weight are a PID controller, Data Collector and an Interaction component. In our setup the PID Controller substitutes the behavior of the hardware controller and controls the robot by joint torques. Therefore, the input of the PID controller component takes the desired target joint configuration and moves the robot by applying torques on the robot joints.

The Data Collector component is switched on while collecting sample data from the simulated UR5 robot in Gazebo. During this sampling, the mapping between joint torques and the corresponding joint configuration is recorded.

Further, the interaction component provides the system with the trained equilibrium model via its port. A steady comparison of joint torques in each joint configuration with the provided torques from the Interaction component allows the compliant control mode to follow external forces.

The presented framework architecture allows to integrate models and machine learning algorithms for physical simulation. Sampling the data is possible with less effort than on a real robot and could be done on several machines simultaneously. The sampled data, e.g. the joint torques, are compared and checked towards plausibility to UR5 joint torques on the real robot. Further, sampling data in simulation avoids safety issues like collisions.

Experimental Setup

For the evaluation of the proposed system, a training data set was recorded for the estimation of the inverse equilibrium model. A distinct test data set is used for evaluation of the quality of the inverse equilibrium model. As argued in Section 5.3.1, the quality of the inverse equilibrium model influences the threshold for the posture update and therefore interaction quality. The better the model approximation of the equilibrium states of the UR5 robot, the lower the threshold T_{th} and consequently

[3] *Gazebo Robot Simulation Tool* - http://gazebosim.org

[4] *Orocos - Open Robot Control Software* - http://www.orocos.org/

[5] *Orocos RTT - Real-Time Toolkit* - http://www.orocos.org/rtt

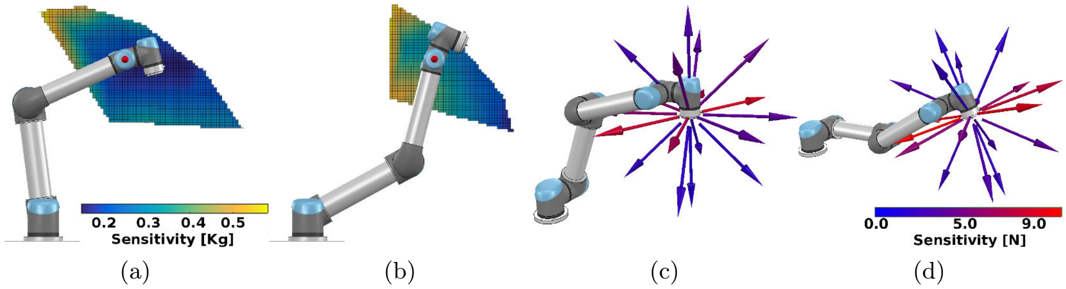


Figure 5.22: Visualization of the sensitivity of posture updates to external forces. The sensitivity to loads at the end effector is shown for postures with positive (a) and negative (b) angles of q_3. The sensitivity is shown for posture changes of q_2 and q_3 inside the workspace. Depending on the posture various sensitivities are achieved. The sensitivity to directed forces for two sample configurations are shown in (c) and (d), due to the redundancy of the robot, different sensitivities for the same end effector position can be achieved.

the higher is the sensitivity of the robot to external forces. In case T_{th} is chosen to low, system noise triggers a position update which results in undesired drifts of the robot. The following experiments aim at the interaction quality by analyzing the sensitivity of the robot in the real application, i.e. simulation. The evaluation of the generalization capabilities for fixed load configurations is presented in Section 5.3.4. Further work that aims at variable load configuration is presented in [Balayn et al., 2016].

Acquisition of the Data Set To ensure data recording in an equilibrium state, the robot's positions and joint torques are measured 3.5 seconds after the joint command is sent in order to reach torque stabilization. The recording constitutes random positions that are chosen in specific intervals inside the robot's workspace. Overall $N_{\text{tr}} = 15625$ random positions are collected, as a grid with 5 random postures per joint are recorded and the robot has 6-DOF ($5^6 = 15625$). Joint positions are joint angles measured in radians and joint torques are measured in Newton meters. The acquired data set is split randomized into equally sized training and test sets. As shown in Section 5.3.3 the real robot has to operate in a constrained workspace, therefore the environment was modeled in simulation and only postures that do not collide with the environment are added to the data set.

5.3.4 Model Learning

Mapping of Joint Angles and Joint Torques To find a sufficient parameterization for model learning, a grid search with five-fold cross validation was conducted. The results of the grid search are shown in Appendix A.2. For the implementation of the inverse equilibrium model, the Extreme Learning Machine (ELM, discussed in Section 2.2.2) comprises $\mathbf{q} \in \mathbb{R}^{I=6}$ input, $\mathbf{h} \in \mathbb{R}^{N_\mathrm{H}}$ hidden, and $\hat{\boldsymbol{\tau}} \in \mathbb{R}^{O=6}$ output neurons. The input is connected to the hidden layer by the input matrix $\mathbf{W}^{\mathbf{inp}} \in \mathbb{R}^{N_\mathrm{H} \times I}$. The read-out matrix is given by $W^{\mathbf{out}} \in \mathbb{R}^{O \times N_\mathrm{H}}$. For input $\mathbf{q}$, the output of neuron o is computed by

$$\hat{\boldsymbol{\tau}}_o(\mathbf{q}) = \sum_{j=1}^{N_\mathrm{H}} \mathbf{w}^{\mathbf{out}}_{oj} \sigma\Big(\sum_{k=1}^{I} \mathbf{w}^{\mathbf{inp}}_{jk} \mathbf{q}_k + b_j\Big) \ , \qquad (5.13)$$

where b_j is the bias for neuron j, and $\sigma(x) = (1 + e^{-x})^{-1}$ the logistic activation function. The components of the input matrix $W^{\mathbf{inp}}$ and the biases b_j are drawn from a random distribution and remain fixed after initialization. The inputs are the current joint angle configurations of the robot and target outputs are the observed torques for each joint.

The data set for training is given by $\mathcal{D} = (A, T) = (\alpha^k, \tau^k)$ with $k = 1 \ldots N_\mathrm{tr}$, where N_tr is the number of training samples. $A \in \mathbb{R}^{I \times N_\mathrm{tr}}$ is the collection of angles, and $T \in \mathbb{R}^{O \times N_\mathrm{tr}}$ is the matrix of target torques for all N_tr samples. Supervised learning is restricted to the read-out weights $W^{\mathbf{out}}$ and accomplished by solving ridge regression. The grid search resulted in the parameterization $N_\mathrm{H} = 500$ hidden neurons and a regularization of $\gamma = 10^{-5}$.

Threshold Estimation Threshold T_th (Equation 5.2) allows an update of the robot posture during simulation. For identification of the threshold, the robot is brought to an equilibrium state at random configurations. The torque differences between the model prediction and the real measurement for each joint are estimated. For extreme configurations in which the robot arm is approximately horizontal or vertical, these prediction errors increase significantly. As the robot should keep a stable position in its whole workspace, the maximum error of each joint is chosen as a joint specific threshold.

Evaluation by Visualization of Sensitivity To evaluate the quality of the interaction, the robot approaches several fixed positions. For evaluation, the additional payload of the robot is measured from which the robot starts updating its posture, as shown in Figure 5.22a & 5.22b. Since this evaluation covers only forces in the direction of the gravity, an additional evaluation with directional forces at the end effector is performed, see Figure 5.22c & 5.22d. The results support hypothesis **H5.1** and show the applicability of an equilibrium based interactive control mode for compliant light-weight robots.

5.4 Discussion

This chapter demonstrated the applicability of learned inverse equilibrium models to improve the low-level control of highly compliant actuators. The proposed hybrid control methods apply classical control concepts in combination with an inverse equilibrium model of the robot. Experimental evaluation shows improved control in terms of tracking error and response time and supports hypothesis **H5.1**. Based on the introduced inverse equilibrium model a compliant interaction mode was proposed. A mismatch of the predicted control signals and the measured control signals of the low-level control is used to trigger an adaptation to external forces as addressed by hypothesis **H5.2**. It allows to perform human-robot-interaction like kinesthetic teaching as demonstrated by the example of an apple picking task.

Chapter 6

Parameterized Skills for Control of Complex Robots

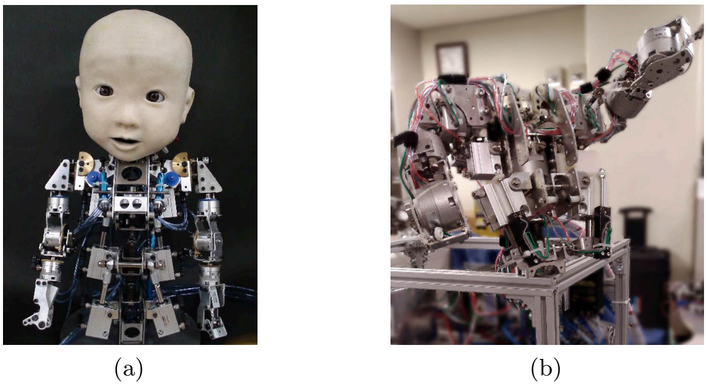
Chapter Overview *This chapter presents an argumentation for task specific generalization of forward signals that support the execution of parameterized policies by the feedback controller. The first part of this chapter evaluates the task specific generalization of forward signals under the assumption that the required kinematic representation, i.e. joint angle trajectories, is available. Experimental evaluation is performed in simulation of a compliant 2-DOF arm and a trajectory tracking task on a 6-DOF pneumatically driven humanoid robot child. The second part of this chapter presents the combination of the learning of forward signals and the generalization of joint angle trajectories for high-level skill learning. Evaluation is performed on a complex scenario that involves kinesthetic teaching, control of a pneumatically actuated robot and dynamic interaction with the environment.*

This Chapter is Partially Based on:

- Queißer, J. F., H. Ishihara, B. Hammer, J. J. Steil, and M. Asada 2018. Skill memories for parameterized dynamic action primitives on the pneumatically driven humanoid robot child affetto. In *Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Tokyo, Japan. IEEE

6.1 Primitive Based Dynamics Representation

As motivated in Chapter 1, modern robot applications often require skill learning that covers task variability. Ijspeert et al. [2013] propose models for action generation based on dynamic motion primitives and perceptual coupling. Further work extends this idea and introduces parameterized skills to perform a generalization of action primitives based on a high-level task description [Kober and Peters, 2010; Silva et al.,



(a) (b)

Figure 6.1: Affetto robot, (a) upper body and internal structure as presented in [Ishihara et al., 2011; Ishihara and Asada, 2015]. (b) Experimental setup. Further information on the robotic system is presented in Section 5.2.

2012; Kober et al., 2012; Baranes and Oudeyer, 2013; Mülling et al., 2013; Reinhart and Steil, 2014; Silva et al., 2014; Queißer et al., 2016].

As previously discussed in Section 2.1 & 5.1, interactive robots that incorporate robust pneumatic actuators have received more attention for real-world applications. In addition to their inherent compliance, a lower susceptibility to overheat and an easy combination with light-weight backdrivable transmission systems, such like proposed by Whitney et al. [2014], is possible. This is important, because the risk analysis of head injuries on collision with robotic actuators by Zinn et al. [2004] shows that one way to lower the risk of injury is the reduction of the inertia of the moving parts of the robot. A further option to enhance safety is a decrease of the stiffness of the actuator.

Unfortunately, the control of pneumatically actuated robots is impeded by delays, friction and complex dynamics. The application of pneumatic robots in interactive scenarios is confronted with additional challenges, like variable configurations of the robot or unmodeled interaction forces. To deal with the aforementioned challenges, the complete dynamics of the robot and the interaction is required for classical model based control approaches. In addition to a parameterization by external factors, the dynamics may evolve over time due to e.g. changing material properties caused by wear-and-tear or task demands. Modeling these properties is difficult or sometimes not possible at all and does not permit a reliable control of the robot system.

This chapter presents an extension of the concept of parameterized skills to generalize for additional feed-forward signals that represent complex dynamic properties and reduce the tracking error of the low-level controller. In comparison to classical approaches that estimate the complete inverse dynamics model of the robot [Kawato et al., 1988; Nguyen-Tuong and Peters, 2011] or hybrid approaches [Nguyen-Tuong and Peters, 2010; Romeres et al., 2016; Reinhart et al., 2017b] that incorporate learning, the proposed approach focuses on representations based on action primitives.

Therefore, it combines kinematic representations with the concept of feed-forward signal generation of the servo theory of the motor cortex [Schweighofer et al., 1998; Kawato et al., 1987; Graziano, 2015b]. For a given parameterization of the task, the parameterized skill (PS) is supposed to estimate a solution in terms of joint angle trajectories that fulfill the task (as demonstrated previously, e.g. Chapter 3) and an associated feed-forward signal that minimizes the tracking error of the joint controller. This allows to shift the complexity from learning the complex dynamics of the robot to task related primitives. In comparison of this work to the torque primitives for impedance control, proposed in [Petrič et al., 2015], a continuous generalization of forward signals based on a high-level task parameterization is performed.

The experimental platform is the Affetto robot [Ishihara et al., 2011], introduced in Chapter 5, which is a pneumatically actuated humanoid with a large number of antagonistically controlled joints. The robot Affetto does not support direct torque control and does not provide dynamics models for reliable joint control. The previously presented model for an inverse equilibrium based controller (Section 5.3.2) improved the tracking performance, but due to the high compliance of the system, interaction and tasks that require a high precision are hardly feasible. Thus, high-level skill learning suffers from the high task complexity as well as delays and dynamic effects caused by the pneumatic actuation. Note, the proposed method to encode task-related feed-forward signals is not limited to pneumatically actuated robots. It is particularly interesting for all robots that are difficult to control by classical control schemes due to their complexity, like e.g. tendon driven actuators or soft robots. The contribution presented in this chapter is an extension of online learning of a parameterized skills for trajectory representations [Silva et al., 2012; Baranes and Oudeyer, 2013; Reinhart and Steil, 2014; Silva et al., 2014; Queißer et al., 2016] to incorporate an additional dynamics representation of highly compliant pneumatic robot systems.

An experimental evaluation of the proposed approach shows an enhanced quality of the control of a simulated compliant 2-DOF planar arm and demonstrates the scalability to a complex real 6-DOF robot system. As in the previously presented work for kinematic representations Chapter 3, a *bootstrapping* process is investigated that results in an acceleration of the optimization process as more training samples have been consolidated by the memory.

The work introduced in this chapter extends the parameterized skill [Queißer et al., 2016] as presented in Chapter 3 and its contribution aims at the experimental verification of the following hypotheses:

H6.1) Motion primitive based generalization of forward signals by the parameterized skill support low-level control of complex robots without the need of an explicit dynamics model. (Section 6.2)

H6.2) The task related generalization of forward signals in combination with a parameterized representation of movements allows to acquire complex skills on a highly compliant robot. (Section 6.4)

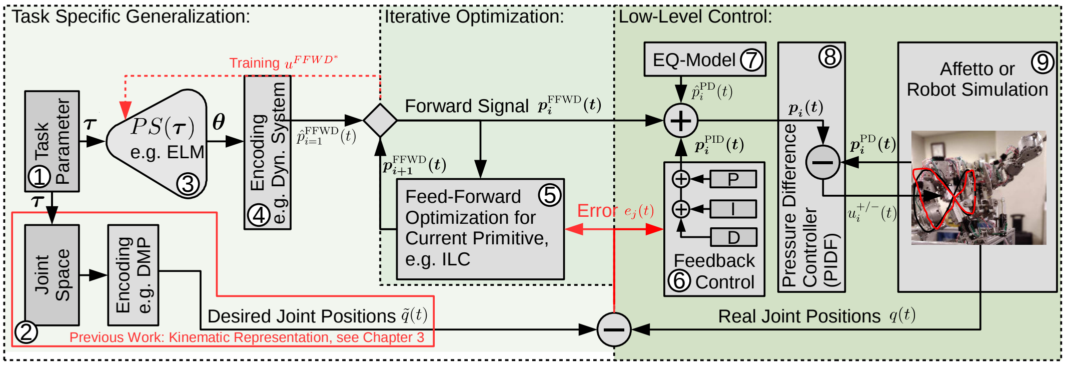

Task Specific Generalization:
Iterative Optimization:
Low-Level Control:
Training u^{FFWD^*}
Task Parameter ①
$\boldsymbol{\tau}$
$PS(\boldsymbol{\tau})$ e.g. ELM ③
$\boldsymbol{\theta}$
Encoding e.g. Dyn. System ④
$\hat{p}_{i=1}^{\mathrm{FFWD}}(t)$
Forward Signal $\boldsymbol{p}_i^{\mathrm{FFWD}}(\boldsymbol{t})$
$\boldsymbol{p}_{i+1}^{\mathrm{FFWD}}(\boldsymbol{t})$
Feed-Forward Optimization for Current Primitive, e.g. ILC ⑤
Error $e_j(t)$
Joint Space ②
Encoding e.g. DMP
Desired Joint Positions $\tilde{q}(t)$
Previous Work: Kinematic Representation, see Chapter 3
EQ-Model ⑦
$\hat{p}_i^{\mathrm{PD}}(t)$
$\boldsymbol{p}_i^{\mathrm{PID}}(\boldsymbol{t})$
P
I
D
Feedback ⑥ Control
⑧
$\boldsymbol{p}_i(\boldsymbol{t})$
Pressure Difference Controller (PIDF)
$\boldsymbol{p}_i^{\mathrm{PD}}(\boldsymbol{t})$
$u_i^{+/-}(t)$
Affetto or Robot Simulation ⑨
Real Joint Positions $q(t)$

Figure 6.2: System overview of the proposed skill learning framework. The parameterized skill PS($\boldsymbol{\tau}$) is the core component and mediates between high-level task parameters and feed-forward signals that represent the dynamic properties of the robot system. Background color indicates functional grouping and the nested loop structure of task parameterization, feed-forward signal optimization and primitive execution.

6.2 Parameterized Skills for Dynamic Action Primitives

Previous work of Chapter 3, introduced parameterized skills as a mapping from task parameterizations to motion primitives. This allows for generalization of actions, i.e. joint angle trajectories encoded by DMPs, for new task configurations and goals [Queißer et al., 2016]. Actions are optimized with respect to a reward function by black-box optimization and used for incremental training of the parameterized skill. For a given task such as 10-DOF arm point reaching, a parameterized skill is able to generalize to adequate actions for new parameterizations (i.e. via-point positions). If the parameterized skill generalizes, but is not successful, the optimizer is used to solve the task. Successfully optimized tasks are used as training data for the parameterized skill and successive optimizations benefit from an improved initialization.

This results in a process that is called *bootstrapping*: the more solutions have been solved, the less rollouts are required for a new optimization. Chapter 3 showed that this leads to a significant speed up of the exploration of the parameterized skill.

For the extended work of this section, it is expected that the generalization of joint trajectories for task parameterizations is already available. To extend the skill learning framework for kinematic representations, the following experiments train parameterized skills to generalize for forward signals that represent the dynamics of the robot and its environment. An overview of the skill learning framework and a differentiation between the kinematic and dynamic representation is presented in Section 2.2. Thus, the parameterized skill generalizes for policy parameterizations that are encoded into forward signals to support the feedback controller in execution of the parameterized target trajectory. This work also constitutes a first step towards the generation of complex dynamic motions, since action primitives can be mixed or sequenced. Training samples are gathered by iterative optimization of the initial guess of the parameterized skill. The experiments evaluate the generalization capabilities of the parameterized skill for forward signals that reduce the tracking error of the feedback controller as well as the iterative optimization of forward signals and online learning.

Figure 6.2 shows the structure of the proposed learning framework: Target trajectories in relation to the task parameterization (Figure 6.2-①) are assumed to be given, as highlighted in red in Figure 6.2-②. The generalization for feed-forward signals $p_{i=1}^{\mathrm{FFWD}}(t)$ for the first iteration $i = 1$ is performed by the parameterized skill $PS(\boldsymbol{\tau})$ (Figure 6.2-③) and its encoding (Figure 6.2-④). Iterative optimization of the generalized feed-forward signal $p_{i+1}^{\mathrm{FFWD}}(t)$ for one task instance (defined by $\boldsymbol{\tau}$) is given by Fig. 6.2-⑤. Optimization is performed until convergence of the tracking error has been achieved. The feed-forward signal giving the lowest tracking error $p^{FFWD^*}(t)$ is used as training target for an incremental update of $PS(\boldsymbol{\tau})$. For action execution, a feedback controller (Figure 6.2-⑥) estimates a control signal $p_i^{\mathrm{PID}}(t)$ based on the current tracking error $e_i(t)$. The utilized low-level controller is the PIDF_EQ_I_RESET controller, as it shows the best performance on the robot platform. Details on the low-level controller are presented in Section 5.3.2. The additional equilibrium model based forward signal is represented by $\hat{p}_i^{\mathrm{PD}}$ as shown in Figure 6.2-⑦. The resulting signal that is processed by the outer loop of the PIDF controller is given by $p_i(t) = p_i^{\mathrm{PID}}(t) + p_i^{\mathrm{FFWD}}(t) + \hat{p}_i^{\mathrm{PD}}$.

The parameterized skill does not estimate the complete inverse dynamics of the robot system and its environment, as performed in case of classical robot control applications for estimation of $p_i^{\mathrm{FFWD}}(t)$. The generalization of optimized p_i^{FFWD} is based on the high-level task parameterization and is supposed to support the feedback controller.

In the case of the Affetto robot, it is not possible to directly command joint torques or accelerations. To abstract the antagonistic control signals that represent the opening of the valves of the pneumatic chambers, the PIDF controller [Todorov et al., 2010] is utilized as shown in Figure 6.2-⑧. Further, the low-level controller incorporates an additional equilibrium model, as discussed in Section 5.3.2, to enhance the precision of the system. This allows to operate with $u(t)$ in the domain of desired pressure differences that correlate to torques at the end effector (Figure 6.2-⑨). The overall system incorporates three nested loops: 1) Generalization of forward signals and the respective joint angle trajectories for each new task instance; 2) Iterative optimization of generalized forward signals; 3) Execution of the joint trajectory by the low-level controller. A more detailed view on the loop structure of the skill learning framework is presented in Section 2.2.

A crucial requirement for the estimation of optimized feed-forward signals is the repeatability of the generated movements of the robot. As investigated in [Todorov et al., 2010] for a humanoid robot with comparable air valves and actuation principle, resulting end effector trajectories showed proper repeatability under multiple executions of identical controller signals. Further, the system is faced with a multi-modal representation: The parameterization of the task will affect the desired trajectory as well as the optimal feed-forward signal, e.g. caused by different loads at the end effector, variable stiffness of the actuator or changing trajectory durations. The evaluation metric is the generalization performance of the parameterized skill for feed-forward signals of unseen task parameterizations. It is expected that the more training samples have been presented to the parameterized skill, the better is the generalized feed-forward signal. Consequently, a gradually increasing tracking performance as well as a reduced number of required optimization steps to achieve convergence of minimizing the tracking error of the system is expected as well.

6.2.1 Component & Task Selection

In the following, the chosen signal representation, the algorithm for feed-forward signal optimization, the selected learning method and the task variability are introduced. The component selection is closely related to the previously presented bootstrapping experiments in Section 3.2.1.

a) **Feed-Forward Signal Representation:**
The proposed method does not rely on a specific type of policy representation, i.e. compact representation and encoding of forward signals to support the execution of motion primitives. Many methods for compact temporal signal representation have been proposed, e.g. based on Gaussian Mixture Models (GMM) [Günter et al., 2007] or Neural Imprinted Vector Fields [Lemme et al., 2014], as discussed in Section 2.2.2. The presented work relies on a dynamical system representation based on Dynamic Motion Primitives (DMP) [Ijspeert et al., 2013], because they are widely used in the field of motion generation and show good task related generalization capabilities. DMPs for point-to-point motions are based on a dynamical point attractor system. For encoding of feed-forward signals as in Fig. 6.2-④, a variant without scaling invariance is implemented. Feed-forward signal $u_{j=1}^{\text{FFWD}}(t)$ as well as its velocity and acceleration profiles are defined as

$$\ddot{p}_{j=1}^{\text{FFWD}} = k_S(g-u) - k_D\dot{p}_{j=1}^{\text{FFWD}} + f_{\text{FFWD}}(x,\boldsymbol{\theta}) \tag{6.1}$$

The canonical system is typically as a linear decay and the disturbance f_{FFWD} is defined as motivated in Section 3.2.1. For the experiments in this chapter, the number of Gaussians set to $K = 20$ per DOF for the feed-forward signal representation. The DMP is parameterized by the mixing coefficients $\boldsymbol{\theta}_k$, generalized by the parameterized skill. Fixed variances $\boldsymbol{V}_k$ and a fixed distribution of centers $\boldsymbol{C}_k$ as in [Ijspeert et al., 2013; Reinhart and Steil, 2015] are assumed. The representation of the joint angle trajectories is performed in the same way as discussed in Section 3.2.1.

b) **Selection of Feed-Forward Signal Optimization Algorithm:**
For optimization of feed-forward signals encoded by policy parameters $\boldsymbol{\theta}$ given

a task parameterization $\boldsymbol{\tau}$, Iterative Learning Control (ILC, [Arimoto et al., 1984; Longman, 1998; Norrloff and Gunnarsson, 2002]) is applied. Integration into the framework is shown in Figure 6.2-⑤. ILC is a method for optimizing control signals and was initially proposed as a solely feed-forward approach. Application in combination with feedback control was demonstrated as well in [Roover and Bosgra, 2000; Bristow et al., 2006]. A successive observation and update of the feed-forward signal leads to a reduction of the tracking error and thereby to a lower feedback controller response. An illustration of the working principle is shown in Section 2.2.2. ILC is widely used in industrial application areas, e.g. for enhancing positioning precision of machines [Chen and Hwang, 2005; Kim and Kim, 1996]. A PD-Type learning function was used for the presented experiments [Bristow et al., 2006]: the feed-forward signal is updated based on a proportional (P) and derivative (D) gain of the current error. ILC is based on a Q-Filter and learning function L. A low-pass filter Q suppresses high frequency learning and contributes to the stability of ILC. Further details can be found in the discussion in Section 2.2.2b and in Figure 2.6. In this case, the Q-filter is given by the representation of the feed-forward signal as the parameterization of a dynamical systems representation (inherent smoothing), additionally a Gaussian filter is applied on the error signal of the joint controller. Iterative adaptation including the update law L of the forward signal is defined as

$$u_{i+1}^{\text{FFWD}}(t) = u_i^{\text{FFWD}}(t) + \underbrace{k_P e_i(t+d) + k_D\left[e_i(t+d+1) - e_i(t+d)\right]}_{\text{Update Law } L(e_i(t))}, \tag{6.2}$$

for iteration i, proportional factor k_P, derivative factor k_D and system delay d. The error $e_i(t)$ over time t is defined by the difference between target joint angles $\tilde{q}_i(t)$ and real joint angles of the current iteration $q_i(t)$: $e_i(t) = \tilde{q}_i(t) - q_i(t)$. For each joint an independent ILC is executed. Due to the high compliance of the application and the pneumatic actuation principle, long and varying temporal delays between the control signal and a response of the actuator can be expected. Therefore, the current temporal delay d of the system depends on the estimation of the time shift with the minimum error between the target and the actuator response: $\min\limits_{d} \frac{1}{T} \sum_t^T ||\tilde{\boldsymbol{q}}(t) - \boldsymbol{q}_j(t+d)||$.

c) **Selection of Learning Algorithm:**

To allow the comparison of the methods that are proposed in this chapter to the bootstrapping of parameterized skills as presented in Chapter 3, the learner configuration was kept unchanged. For learning of parameterized skills PS($\boldsymbol{\tau}$) an incremental variant of the Extreme Learning Machine (ELM, [Huang et al., 2006]) was implemented as discussed in Section 3.2.1. As before, hidden Layer size was set to $N_{\text{H}} = 50$ for generalization in joint space. Linear regression is applied on a random projection of the input $\mathbf{W}^{inp} \in \mathbb{R}^{N_{\text{H}} \times E}$, a nonlinear transformation $\sigma(x) = (1 + e^{-x})^{-1}$ and a linear output transformation $\mathbf{W}^{out} \in$

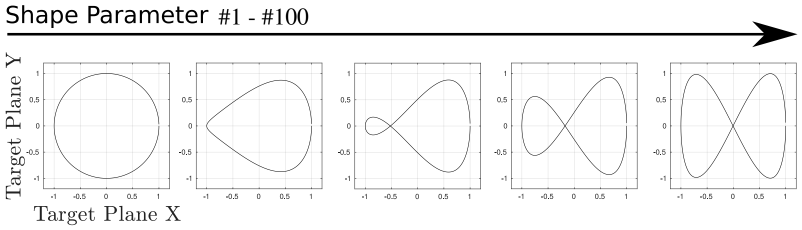


Figure 6.3: Shape variation at end effector that is used for evaluation.

$\mathbb{R}^{F \times N_\text{H}}$ that can be updated by incremental least squares algorithms. A more detailed discussion on the learning method and parameter estimation of the readout weights is presented in Section 2.2.2.

d) Selection of Parameterized Task:
For the experiments an evaluation of parameterized 2D end effector tracking tasks is performed, as shown in Figure 6.3. Additionally, the end effector loads are varied in simulation as well as the overall duration for the real robot of the action primitives. As mentioned before the learning of the feed-forward signals assumes that the joint angle trajectories are predefined.

6.3 Evaluation of the Dynamics Representation

In the following, the applicability of the proposed bootstrapping algorithm is presented. Therefore, two scenarios have been designed to test the bootstrapping of parameterized skill according to the method presented in Section 6.2 for the representation of forward signals.

6.3.1 2-DOF Planar Arm Task

The first experiment was performed in simulation of a 2-DOF planar arm. The simulated compliant planar arm was modeled in the simulation environment VREP [Rohmer et al., 2013]. To simulate a highly compliant actuator, two simulated joints are added for each DOF of the robot. One joint is supposed to simulate a spring-damper system and the other joint is controlled in torque mode. The simulation of the dynamics was performed by the *Newton Dynamics* engine with a temporal resolution of 20ms. Each DOF is driven by a feedback controller that considers the error between the target joint angle and the measured joint angle. The measured joint angle is given by the combination of the actuated and the compliant joints, as shown in Figure 6.7b. Based on this error, the feedback controller scheme results in a control signal for the actuated joint. In addition, the parameterized skill provides a forward signal so that the final control of the actuated joint is based on the sum of the feedback controller and the forward signal. As presented in Section 6.2, the task is parameterized by the shape of the end effector trajectory and appropriate joint angle trajectories are estimated by the inverse kinematic solver of VREP. As a second dimension of the parameterization of the task, the weight of a load that is attached to the end effector of the robot is varied. The evaluation of the generalization properties of optimized forward signals for single instances is shown in Figure 6.3. The tracking performance of the PID controller with a zero forward signal (baseline) is compared to three conditions in which the low-level controller is supported by forward signals gathered by optimization of ILC for a specific shape parameterization (#1,#50 and #100, see Figure 6.3). The parameters for the iterative ILC update have been estimated as $K = [k_P, k_D] = [0.005, 0.04]$ by manual tuning with a Gaussian window filter size of 100 timesteps. As it can be seen in Figure 6.4, the tracking error is much lower for the shape parameterizations if the forward signal is optimized for this specific shape (colored vertical bars). The more the shape deviates from the shape for which the forward signal was optimized the higher the tracking error, since the used feed-forward signal was not optimized for the selected shape. If the forward signal was optimized for a shape that strongly deviates from the shape used for optimization, the tracking error of the controller that utilizes the forward signal can be higher compared to the case in which no forward signal is used. In this case, the forward signal perturbs the trajectory tracking and is not beneficial for the feedback controller. This experiment shows that the optimized forward signals are beneficial in a local neighborhood of the task parameterization and generalization for task parameterizations is feasible.

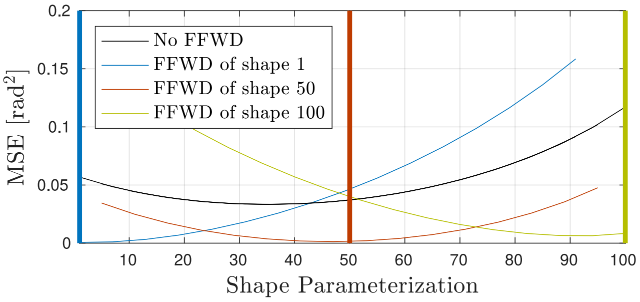
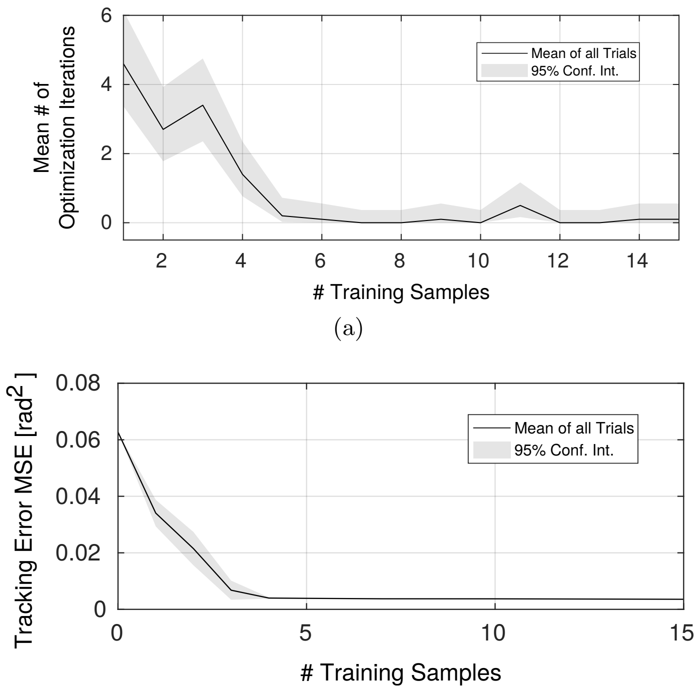


Figure 6.4: Evaluation of generalization capabilities of forward signals with respect to the task parameterization. Resulting tracking error of the 2-DOF arm with zero forward signal (black) is compared to conditions in which the optimized forward signal (FFWD) for a specific shape parameterization is used (#1, #50 and #100).

Based on the aforementioned observations, the evaluation of the generalization capabilities of the parameterized skill is performed in the second experiment. To evaluate the system performance during the presentation of random training tasks, a fixed test set of parameterizations over shapes and load (0-2kg) has been generated. For each iteratively presented training task instance, the generalization of feed-forward signals of the parameterized skill is evaluated. Given this initial feed-forward signal an iterative update of the forward signal by ILC is performed for optimization. Iterations are performed until a convergence criterion of the joint tracking error is fulfilled. Subsequently, the optimized forward signal for the given task instance



(a)

(b)

Figure 6.5: Results of 2-DOF arm experiment. (a) The mean number of rollouts that are necessary for optimization by ILC until convergence and (b) the tracking error for parameterized tasks for forward signals decoded from $\boldsymbol{\theta}_{\mathrm{PS}} = \mathrm{PS}(\boldsymbol{\tau})$ in relation to the number of presented training samples. Results and confidence intervals are based on ten repeated experiments.

is used as a training sample for the iterative update of the parameterized skill. The evaluation of the generalization capabilities is performed by the estimation of the tracking error on the test set. The results of this procedure are shown in Figure 6.5, the MSE of the trajectory tracking task decreases with an increasing number of presented training task instances. Additionally, it can be observed that the number of iterations that are required to achieve convergence of the ILC for new training tasks decreases as more solutions for tasks have been consolidated by the parameterized skill. This allows for a bootstrapping of the learning process: the more experience the system has in solving task instances the faster it can find solutions for unseen instances. Figure 6.7c-6.7k shows the tracking performance of the end effector for three shape parameterizations as more samples have been presented to the parameterized skill. The results reveal that learning is successful, as the system gradually enhances the precision of the task execution. After the presentation of only two samples a higher variance in the generated samples can be observed, which is caused by the high shape variability of the randomly selected training tasks.

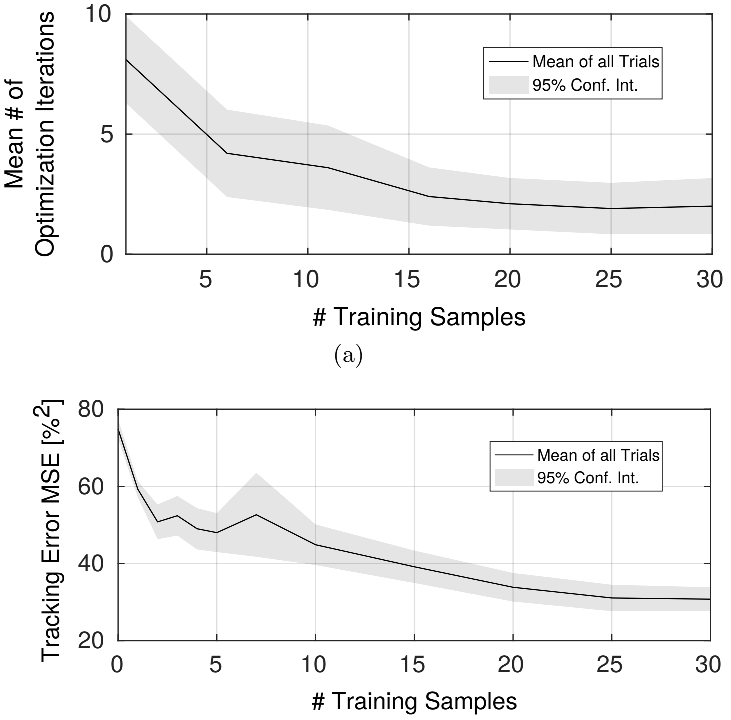
6.3.2 Upper Body Control of the Affetto Robot

The second part of the evaluation is performed on the Affetto robot platform, as shown in Figure 6.1. Further information on the robot platform is presented in Section 5.2. The Affetto is a humanoid robot child that is driven by pneumatic actuators, as introduced in [Ishihara et al., 2011; Ishihara and Asada, 2015]. For the following experiments 6-DOF of 8-DOF (#①-#⑥, see Figure 5.3) of one side of the upper body of the Affetto robot are utilized. The generated joint angle trajectories are parameterized by the shape of the resulting end effector trajectory of the right arm. The remaining 2-DOF are assumed to be optional joints and neglected in the following evaluation. As before, experiments are based on parameterized end effector trajectories as described in Section 6.2, but instead of a load, the duration of the actions is varied (1.6-26.6 seconds) by a second task parameter. As for the 2-DOF experiment a kinematic model and the inverse kinematic solver of the VREP simulator are utilized. It is ensured that the generated joint angle trajectories do not contain multiple solutions of the redundancy resolution and can be represented as parameterized functions. The simulation of the kinematics is shown in Figure 6.8a. The PIDF controller [Todorov et al., 2010] is used as a basis for control of the pneumatically driven joints of the robot and extended according to Section 5.3.2 by an equilibrium model and a reset of the integral component. The controller parameters are optimized by automatic optimization and hand tuning on a test trajectory that includes sine waves and step responses. Further details regarding the low-level control can be found in Section 5.3. A grid search was performed to estimate appropriate parameters for the iterative PD update step of ILC as well as the filter width, as introduced in Section 6.2. The result of the grid search is shown in Figure 6.8b, tracking performance was evaluated for shape parameterization #50. Based on this evaluation the Gaussian window filter with was set to a width of 20 time steps and the update rate factor to $0.75K$. As presented in Figure 6.8b, smaller filter widths or larger step sizes do not result in lower tracking errors but enhance the risk for instabilities during ILC optimization.

For evaluation of the system performance, the same scenario as for the 2-DOF experiment of Section 6.3.1 was selected. The low-level controller for the antagonistic actuators is defined as

$$u_i^+ = k_F(p_i^{\mathrm{PID}} + p_i^{offset} + \hat{p}_i^{\mathrm{PD}}(\mathbf{q}) + \hat{p}_i^{\mathrm{FFWD}} - p_i^{\mathrm{PD}}). \tag{6.3}$$

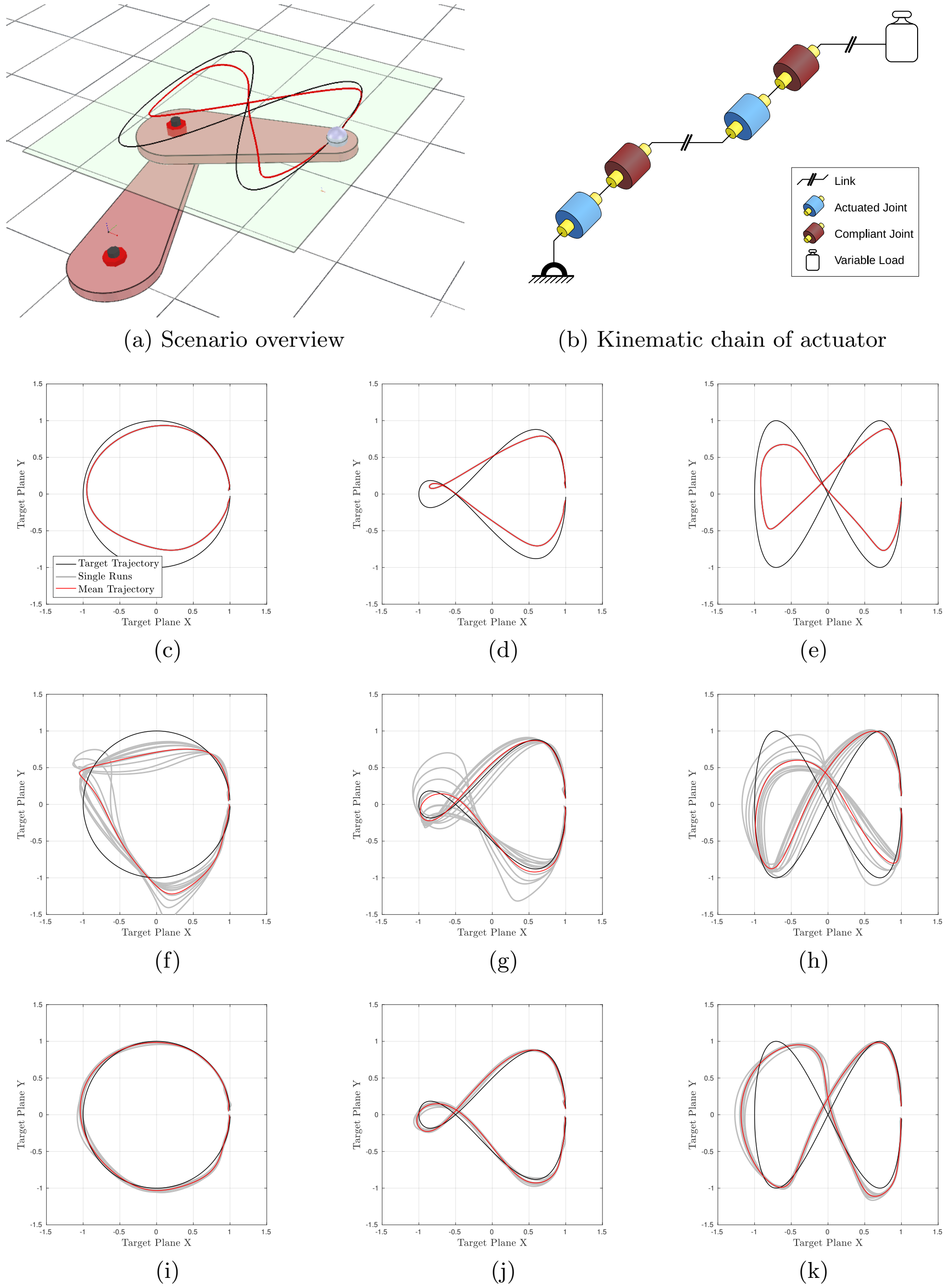
It is based on the PIDF_EQ_I_RESET controller as introduced in Section 5.3.2 and extended by the forward signal $\hat{p}_i^{\mathrm{FFWD}}$. As Figure 6.6 shows, the real robot experiments reproduced similar results as the simulation of the 2-DOF arm. The parameterized skill is able to enhance the generalization incrementally for unseen task parameterizations. The more samples have been used for training of the parameterized skill, the lower the tracking error for the test set. Additionally, it can be seen that the same bootstrapping effect as in the previous experiment occurs, a significant reduction of the required ILC iterations with the gradually enhanced parameterized skill can be observed. As in the previous experiment, the kinematics model was used to visualize the tracking performance of the end effector for three shape parameterizations during presentation of training samples to the parameterized skill, Figure 6.8c-6.8k.



(a)

(b)

Figure 6.6: Results of Affetto experiment. (a) The mean number of rollouts that are necessary for optimization by ILC until convergence and (b) the tracking error for parameterized tasks for forward signals decoded from $\boldsymbol{\theta}_{\mathrm{PS}} = \mathrm{PS}(\boldsymbol{\tau})$ in relation to the number of presented training samples. Results and confidence intervals are based on ten repeated experiments.



(a) Scenario overview

(b) Kinematic chain of actuator

(c)

(d)

(e)

(f)

(g)

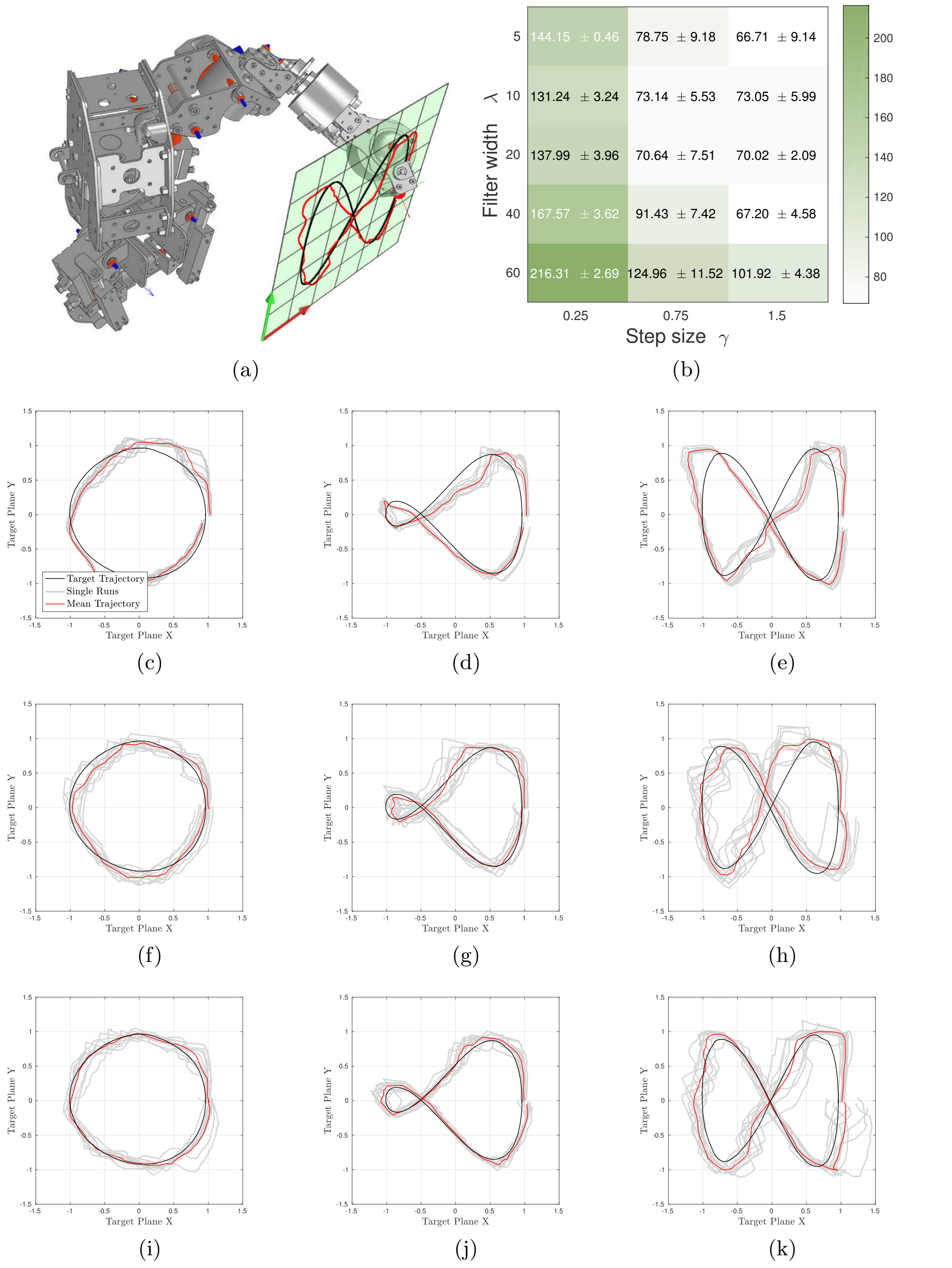
(h)

(i)

(j)

(k)

Figure 6.7: (a) Experimental setup of the compliant 2-DOF arm experiment. Due to the high compliance of the robot, tracking tasks on the 2D target plane (black line) result in perturbed trajectories (red line). (b) Kinematic chain of the simulated actuator. (c-k) Examples of the generalization of PS($\boldsymbol{\tau}$) to unseen tasks. For three shape parameterizations and a fixed load, resulting target trajectories for zero forward signal (c-e), with a parameterized skill trained with two samples (f-h) and with 10 presented training samples (i-k) are shown.



(a) (b)

(c) (d) (e)

(f) (g) (h)

(i) (j) (k)

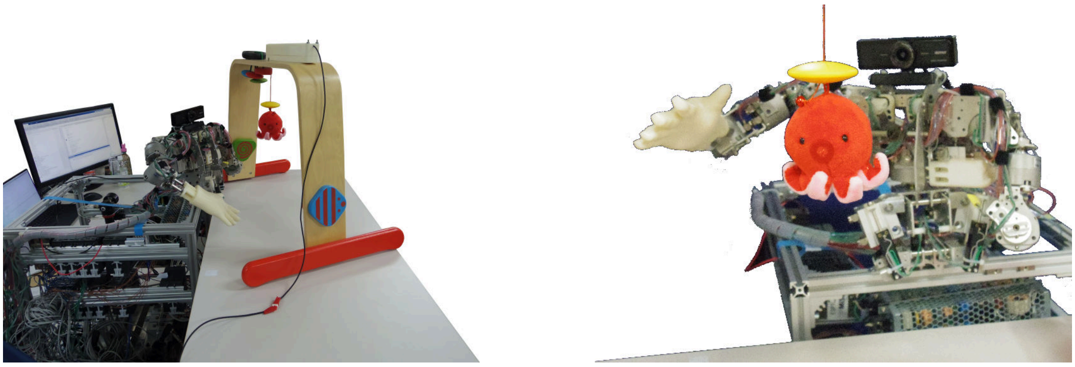
Figure 6.8: (a) Experimental setup of the Affetto experiment. Tracking tasks on the 2D target plane (black line) results in perturbed trajectories (red line). (b) Results of parameter grid search of ILC filter width and step size. Mean minimum reached MSE of three trials and range that includes all trials. (c-k) Examples of the generalization of PS($\boldsymbol{\tau}$) to unseen tasks. For three shape parameterizations and a fixed load, resulting target trajectories for zero forward signal (c-e), with a parameterized skill trained with two samples (f-h) and with 20 presented training samples (i-k) are shown.

6.4 Interaction in Dynamic Environments by Integration of Kinematics and Dynamics

This section aims at the demonstration of the full potential of the proposed parameterized skill framework for a primitive based kinematics and dynamics representation, as proposed in Section 2.2. Therefore, the final scenario requires the complete 8-DOF of the Affetto robot from the wrist to the hand of the right arm, as presented in Figure 5.3. For a precise control of the robot's actuators, the previously proposed PIDF_EQ_I_RESET controller is implemented as introduced in Section 5.3.2. An overview of the scenario arrangement is presented in Figure 6.9. Besides the task parameter that describe the position of the toy in front of the robot, the system is faced with the challenge of a precise control of 8 pneumatic DOF of the robot. Due to the given task in which the robot has to pull down the toy to generate a reward value, the robot has to interact with the environment in a closed loop setup. In the following list summarizes the challenges of the depicted scenario:

- Complex robot: sensory noise, 8-DOF, parallel kinematics, linear and rotary actuators
- High compliance: pneumatic actuation
- Precise and powerful movements: triggering of a spring mechanism is required to fulfill the task
- Parameterized scenario: robot has to generalize for different target positions that induce strong variations of the robot's movements
- Interaction with the environment: object manipulation (elongation of spring mechanism)
- Difficult task: feed-forward signal is required for successful task execution

Scenario Setup The experimental setup simulates a typical scenario in which a child is supposed to play with a toy. The upper body of the Affetto robot is located in front of a baby gym as shown in Figure 6.9a. In the center of the baby gym, a squishy toy is attached to a cable that is connected to a spring mechanism which is mounted on top of the baby gym. The goal in this scenario is to pull down the toy. A closer view on the toy, the attached cable, and the robot is shown in Figure 6.9b. The spring mechanism is equipped with a cable length sensor. By pulling down the squishy toy, the spring mechanism gets triggered. The spring deflects with a ratio of $1/2$ in relation to the length of the cable. Thus, the length sensor measures the deflection of the spring and thereby the distance the toy was pulled down from its initial position. A solid hand is attached as end effector to allow a manipulation of the toy by the robot. The solid hand is equipped with spread-out fingers, thus, the robot can hook the cable of the squishy toy between the fingers to pull it down.



(a) (b)

Figure 6.9: Scenario overview of the interaction scenario

Nevertheless, to successfully hook the squishy toy between the fingers a high precision, synchronization, and coordination of the 8 pneumatic DOF of the robot are required. Besides the difficulty of interaction with an object, the robot has to overcome the strong counterforce of the spring mechanism. The required precision and strength of the movement cannot be handled by the feedback controller as it suffers from compliance and long control delays. This ensures that an additional feed-forward signal for the low-level controller is necessary to be able to fulfill the task. The reward function for evaluation of the success of a performed action is given by

$$R(\boldsymbol{\theta}) = \frac{1}{e^{-10(u_{\text{toy}}-0.5)}}. \tag{6.4}$$

A sigmoid function limits the measured sensor values u_{toy} of the cable length to stay in the interval $[0, 1]$. The distance the toy was pulled down from its initial position is estimated by $\Delta l_{\text{toy}} = 101.6 \cdot {}^{u_{\text{toy}}}\!/_{3.3}$ in centimeters. The baby gym can be freely moved on a table in front the robot. The parameterization for a current task instance is estimated by locating the red colored squishy toy in the center of the camera image. As for the drumming scenario described in Section 3.4.3, the camera is attached to the upper body of the robot.

Experimental Evaluation of Generalization Capabilities The main aim of the experiment is to evaluate the generalization capabilities of the robot to adapt its actions for unseen positions of the target object.

The task of the robot is to trigger the spring mechanism by pulling down the plushy toy that is located in the center of the baby gym. Compared to the previous experiments, successful trajectories for specific positions of the toy are gathered by human demonstrations instead of policy optimization. Kinesthetic teaching is selected for two reasons: on the one hand, it allows to reduce the experimental time significantly as no optimization of the policy has to be executed. On the other hand, optimization by policy search is difficult. Successful optimization of actions would require a further extraction of features of the visual stream during execution. As an example, the reward function (Equation 6.4) does not provide information about the distance of the hand to the toy, therefore an optimization of the movements of the robot is hardly feasible by CMA-ES.

Only successful human demonstrations (reward of the action exceeds a threshold $R(\boldsymbol{\theta}) \geq 0.85$) are used for further processing and training data acquisition. For each human demonstration, the robot performs an iterative optimization of the required forward signals to minimize the tracking error and to reproduce the successful human demonstration. It is assumed that the minimization of the tracking error is intrinsically tied to the maximization of the reward function. It is neglected that the assumption is not met all the time, as discussed in the results section of the performed experiments. For all training targets that consist of a successful human demonstration and the optimized feed-forward signals, the parameterized skill is trained with pairs of $(\boldsymbol{\tau}_i, \boldsymbol{\theta}_i)$ for $i = 1 \dots N_{\text{tr}}$. Up to 12 randomly selected training samples for positions of the toy in the reachable workspace of the robot are presented to the memory. During the incremental training an evaluation for the generalization to a test set of unseen task instances is performed. The test data set consists of six fixed random positions distributed in the reachable task space of the robot for evaluation. All experiments are repeated ten times with random learner initializations. Evaluation is performed based on the reached reward for the unseen task instances. The following paragraphs will introduce the details of each step.

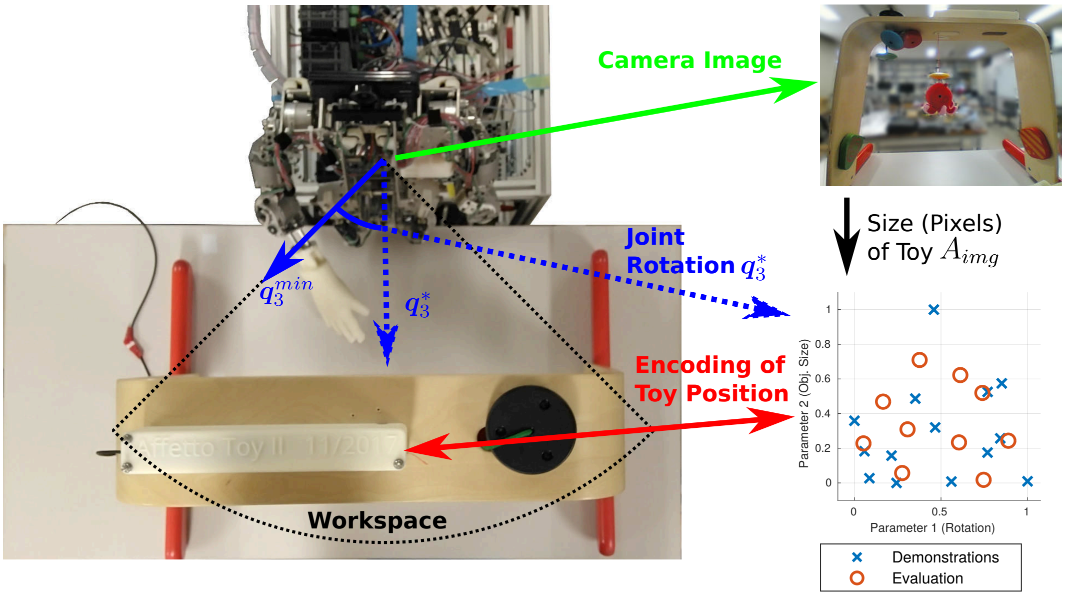


Figure 6.10: Top-down view of the Interaction Scenario setup. The robot is mounted in front of a table. The baby gym with the attached target object can be freely moved on the table.

Acquisition of Training Data set For each of the $N_{\text{tr}} = 12$ training configuration of the target toy, a human tutor presented a successful trajectory (i.e. exceed threshold on reward) by kinesthetic teaching.

The following experiments use the PIDF_EQ_I_RESET controller that incorporates the equilibrium model as well as reset of the integral part, as it showed the best tacking performance. Further details on the low-level controller are discussed in Section 5.3.2.

As previously introduced for the recording of trajectories for the drumming scenario in Section 3.4.3, a predefined initial posture $\boldsymbol{q}^{start}$ is commanded as a target for the joint controller to initiate the teaching mode. After convergence of the robot to the initial posture, the compensation of disturbances by the feedback controller is temporarily deactivated by setting the integral component of the controller to zero $k_I = 0$. It can be expected that the activated equilibrium model of the PIDF_EQ_I_RESET controller compensates for the integral component of the controller for equilibrium states, as the proportional and derivative components have a zero contribution in such cases. A deflection of the robot joint's configuration from $\boldsymbol{q}^{start}$ during the demonstration phase results in a counter force caused by the feedback controller's proportional gain. Thus, the robot tends to move its position back to the initial configuration, as in the case of the drumming scenario in Section 3.4.3. This control scheme results in an impedance control like behavior and supports the demonstrator during kinesthetic teaching. Each trajectory recording is run for $t_{\text{rec}} = 3$ seconds. For each of the 8-DOF of the robot, a DMP with $K = 15$ basis functions represents the joint angle trajectories encoded in $\boldsymbol{\theta}_K$. The kinesthetic teaching of the robot for different positions of the toy is shown in Figure 6.12. During a duration of t_{rec}, the human demonstrator has to: 1) move the robot hand towards the squishy toy; 2) hook up the cable between middle and ring finger; 3) pull the squishy toy down to exceed a reward/length threshold; 4) move the hand upwards to release the toy; 5) return to the initial configuration $\boldsymbol{q}^{start}$. Returning to the initial configuration during demonstration was supported by the controller of the Affetto robot. At the initial position, the demonstrator cannot feel any counter force of the robot, due to the activated equilibrium model of the controller. For the representation of the forward signals $\boldsymbol{\theta}_{\text{D}}$, a dynamical systems representation similar to DMPs was implemented, as introduced in Section 6.2.1. PD-type ILC was executed for 20 iterations to optimize the forward signals that reduce the tracking error. The collected data is added to the training data set, in case the optimized action was classified as successful by the reward function (reward exceeds a certain threshold). The introduction of the ILC update and further discussions are presented in Section 2.2.2 & 6.2.1. The final policy parameterization is defined as $\boldsymbol{\theta} = [\boldsymbol{\theta}_K \, \boldsymbol{\theta}_D] \in \mathbb{R}^{35}$, as introduced in Section 2.2.

The results of the minimization of the tracking error by ILC for the demonstrated trajectories is shown in Figure 6.11a. Since demonstrations represent successful actions, i.e. return a high reward, the resulting reward for the demonstrated actions rises as as more iterations of ILC have been performed, Figure 6.11b. In Appendix A.3,

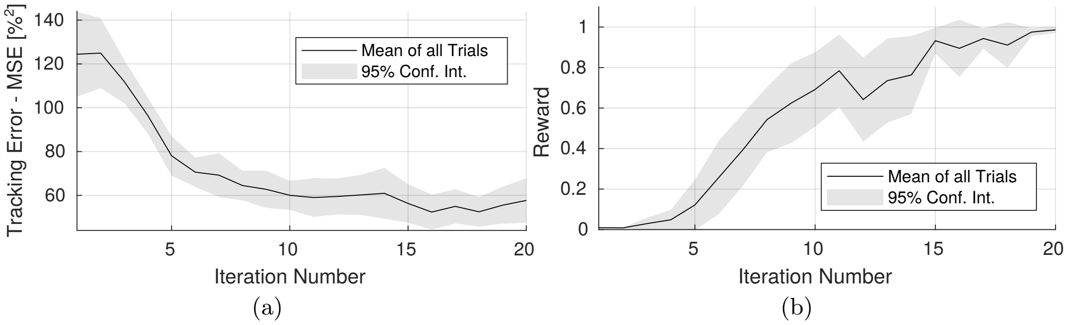


Figure 6.11: Evaluation of the optimization of all $N_{\text{tr}} = 14$ human demonstrations. (a) Tracking error of the reproduction of the training samples during optimization of forward signals. (b) The returned rewards correlate with the cable length.

the single evaluations for each demonstration are presented. Note, that due to the execution of ILC for each joint independently and the complex interaction, a continuously decreasing tracking performance is not guaranteed. As an example, the increasing precision of the joint controller allows to hook the toy, but due to the successful hooking of the toy the robot cannot move its arm downwards due to the counterforce of the spring mechanism. Therefore, the tracking error can temporarily increase until further iterations of ILC compensate for the load of the spring mechanism, as observed in case of Appendix A.3a.

The camera that is attached to the upper body of the robot performs a basic visual search and blob detection of the squishy toy that is attached to the baby gym. The object detection return the horizontal $x_{\text{img}} \in [0, 1]$ and vertical $y_{\text{img}} \in [0, 1]$ position of the center and object size in pixels A_{img}, normalized for reachable positions in the workspace. To estimate the task parameterization, the robot moves to a fixed starting configuration $\boldsymbol{q}^{\text{start}}$ (shown in Figure 6.9a) and centers the toy in the image of the camera by only rotating the upper body orientation by joint q_3.

As soon as the robot has aligned its upper body to the direction of the squishy toy, the size of the pixel area of the toy is calculated. The calculation of the number of pixels that correspond to the red colored toy A_{img} is performed by a simple blob detection on the visual image of the camera. The result is a 2D-vector that represents the location of the squishy toy in relation to the robot $\boldsymbol{\tau} = (A_{\text{img}}, q_3^*)^\top$. As for the drumming scenario, the task parameterization encodes the final rotation of the upper body $3q_3^*$. The estimation of the task parameterization is illustrated in Figure 6.10. The final outcome is a 2D parameterization that represents the position of the drum in relation to the robot.

Finally, the kinesthetic teaching and the optimization of the robot results in the training set $\mathcal{D} = \{(\boldsymbol{\tau}^k, \boldsymbol{\theta}^k) | k = 1, \ldots, N_{\text{tr}}\}$, that is presented in a random order for an incremental training of the parameterized skill.

Further, a per-joint analysis of the dataset for parameterized trajectories and forward signals reveals the high complexity of the task, as shown in Section A.7.

It can be seen that the trajectories as well as control signals vary significantly for generalizations in the workspace. Not only the joint angles vary in relation to the task parameterization, also the controller signals (in particular #2, #4 & #5) indicate that the load of the robot changes between the joints in relation to the task parameterization (i.e. toy position).

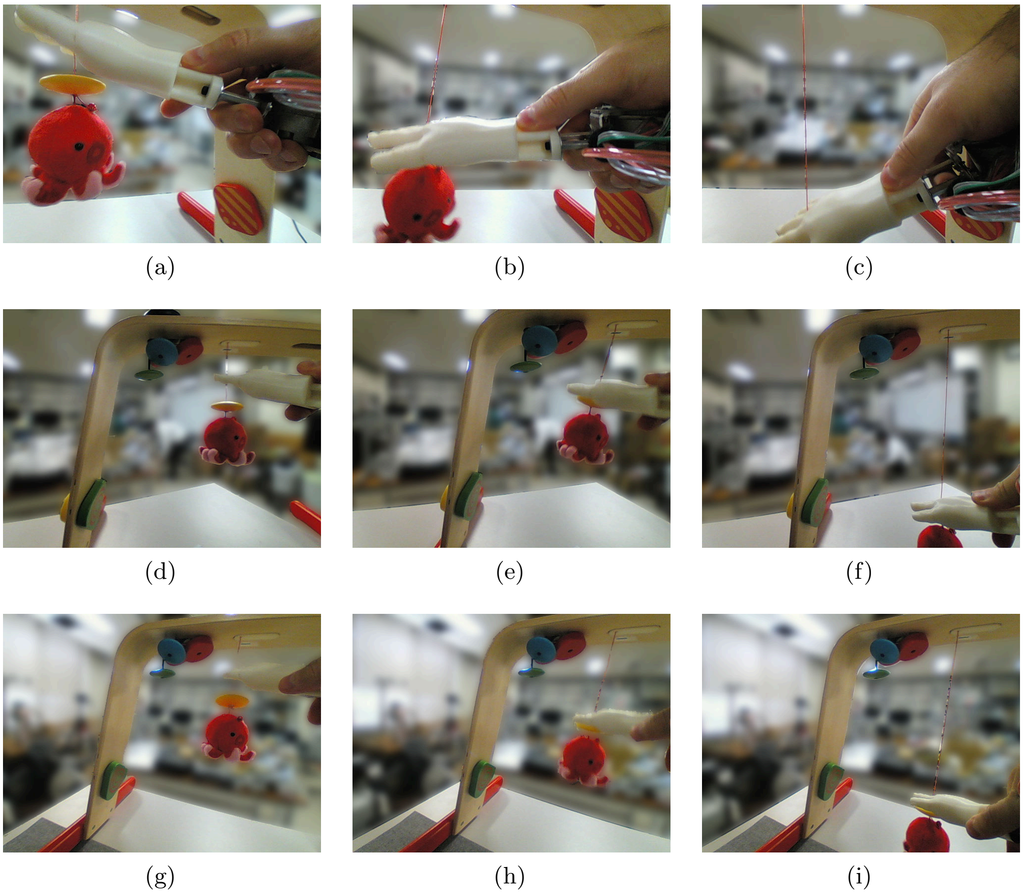
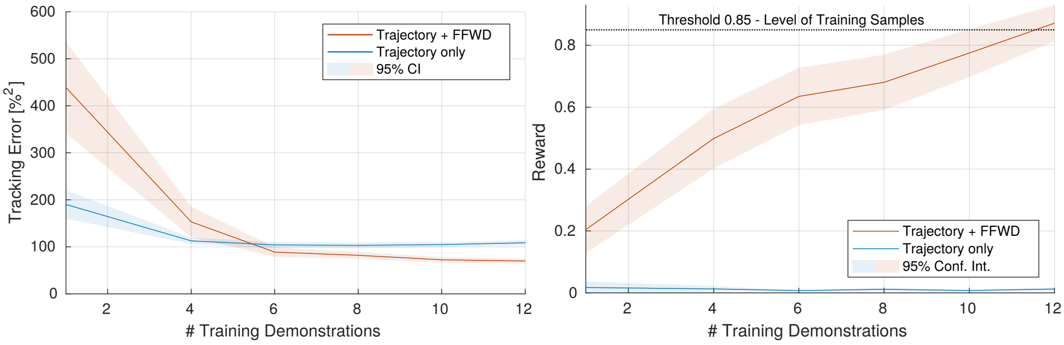

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 6.12: Exemplary human demonstrations. The range of motion patterns for close, medium, and far distances to the robot includes different strategies.

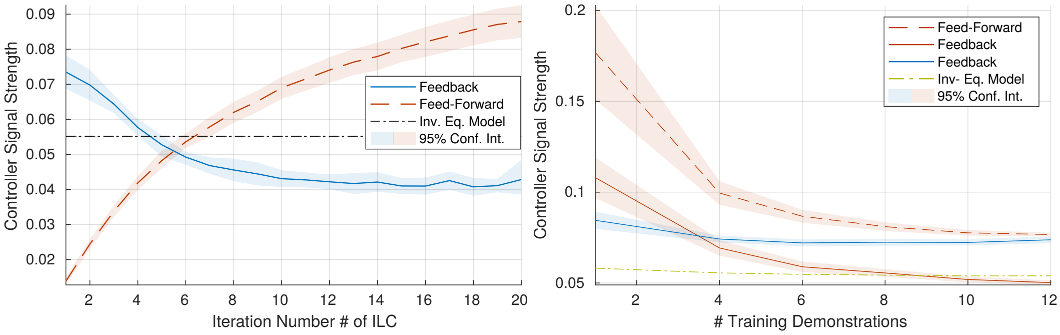
Evaluation of Generalization Capabilities Based on the previously recorded training set, a parameterized skill is trained. The configuration of the learner was depicted in the same way as described in Section 6.2.1. Evaluation of the performance of the parameterized skill is performed by the reward function (Equation 6.4) that assesses how well the robot is able to pull down the squishy toy. The distribution of training and test data in the space of the task parameterization is shown in Figure 6.9a. For evaluation ten unseen random positions have been selected and evaluation has been repeated ten times ($N_{\mathrm{tr}} = 10 \cdot 10$) under random initialization of the training sequence and weights of the parameterized skill. For a comparison to a baseline, the experiment was repeated without the estimation of the feed-forward signals, i.e. $\boldsymbol{\theta}_D = 0$ and thus $\boldsymbol{p}^{\mathrm{FFWD}} = 0$. Without an additional feed-forward signal, the parameterized skill restricts its representation to joint trajectories and does not support the low-level controller. The result of the experiments are presented in Figure 6.13. Figure 6.13a shows the tracking performance of the low-level controller for the generalized joint trajectories of the parameterized skill. It reveals that the generalization of additional feed-forward signals allows to reach a lower tracking error only if more than five training demonstrations have been presented to the memory. In fact the spring mechanism got triggered for all evaluated positions in the test set, exemplary snapshots during solving the test tasks are shown in Section A.8. In the case that no forward signals are represented by the parameterized skill, the precision of the executed actions of the robot is low and the robot fails to hook the toy. Therefore, the robot's actuators do not have to work against the force of the spring mechanism and lower tracking errors can be reached, although the performed action is not successful and the reward is low, as shown in Figure 6.13b. In case the parameterized skill generalizes additional forward signals to improve the tracking error, the robot is able to position the toy between the fingers and pull it down. But pulling down the toy against the spring mechanism can only be handled successfully after the presentation of further demonstrations. It can be seen that as more demonstrations have been consolidated by the system, the success rate of the robot to pull the squishy toy increases. A further evaluation investigated the resulting controller signals during execution of the actions. The resulting magnitudes of the control signals are shown in Figure 6.14. Figure 6.14a shows the feedback and feed-forward components for execution of the training trajectories. The results show that the forward signal (red) $\boldsymbol{p}^{\mathrm{FFWD}}$ becomes stronger as more ILC iterations are performed. Consequently, the feedback signal $\boldsymbol{p}^{\mathrm{PID}}$ becomes lower as less model uncertainties have to be compensated. The contribution of the inverse equilibrium model $\hat{\boldsymbol{p}}^{\mathrm{PD}}(\mathbf{q})$ stays constant as the same actions are performed for all evaluations. Due to the applied PID-controller and the strong proportional gains, similarities between the feedback signal and the tracking error can be identified. As expected, the generalized forward signals, that originate from the optimization by ILC, are able to significantly reduce the magnitude of the feedback for the demonstrations as well as for the generalization for unseen task instances. In case no forward signals are used, a reduction of the controller signals for low numbers of presented demonstrations can be observed as well. This effect occurs due to self-collisions or collisions with the baby gym for low numbers of consolidated demonstrations by the parameterized skill. After the presentation of four demonstrations, no further noticeable reduction of the controller signals can be observed for the system that does not generalize for additional forward signals. Figure 6.14b presents the magnitude of the controller signals for the controller that has no access to generalized forward signals (blue) and the proposed controller of this chapter that combines feedback control $\boldsymbol{p}^{\mathrm{PID}}$ (red, line) with generalized forward signals $\boldsymbol{p}^{\mathrm{FFWD}}$ (red, dashed). The signal magnitude is estimated by the mean of the absolute value of the respective control signals. The results show that the generalization of forward signals is successful and reduces the feedback controller response in comparison to the controller that is limited to feedback.



(a)

(b)

Figure 6.13: Evaluation of the generalization performance of the parameterized skill after presentation of 1-12 of the $N_{\mathrm{tr}} = 14$ human demonstrations. (a) Tracking error and (b) reward values for 10 different task parameterizations and 10 repetitions ($N_{\mathrm{te}} = 100$).



(a)

(b)

Figure 6.14: Feedback controller signal strength in relation to the number of iterations of iterative learning control (ILC). (a) Mean values of $|\boldsymbol{p}^{\mathrm{PID}}|$ (blue), $|\boldsymbol{p}^{\mathrm{FFWD}}|$ (red) and $|\hat{\boldsymbol{p}}^{\mathrm{PD}}(\boldsymbol{q})|$ during iterative optimization of all $N_{\mathrm{tr}} = 14$ human demonstrations. (b) Mean values for $|\boldsymbol{p}^{\mathrm{PID}}|$ (red, line) and $|\boldsymbol{p}^{\mathrm{FFWD}}|$ (red, dashed) in comparison to $|\boldsymbol{p}^{\mathrm{PID}}|$ (blue) of a controller without integration of forward signals. Results based on 10 different task parameterizations and 10 repetitions ($N_{\mathrm{te}} = 100$).

6.5 Discussion

In this chapter examines the applicability of parameterized skills for generalization of feed-forward signals that support the feedback controller on control of highly compliant robots. The presented experiments verify that incremental learning of parameterized skills for representation of forward signals is possible as stated by hypothesis **H6.1**. Incremental learning can significantly reduce the tracking error of the humanoid robot Affetto as well as the number of required optimization iterations for unseen task instances. One of the most fundamental argument throughout this work is that learning is not bound to the complexity of the robot and its environment since the system performs an action/task related generalization. The experiments demonstrated the working principle on a chain of six highly compliant pneumatically actuators without to refer to complex (model based) control strategies that deal e.g. with friction nor time delays. Even under this extreme conditions, it was possible to optimize for a complex task with a low number of rollouts.

Further, the proposed skill learning architecture was evaluated on a complex scenario. The designed task requires interaction with the environment to solve a parameterized task and addresses hypothesis **H6.2**.

Chapter 7

Discussion & Conclusion

The main aim of this thesis is to investigate efficient skill learning that can be applied on highly compliant robotic systems. Therefore, this thesis proposes a novel skill learning framework that was applied on (even though it is not limited to) pneumatically driven robotic systems. The proposed framework is based on earlier research on parameterized skills, a memory structure that generalizes from a high-level task parameterization to robot actions that fulfill given task constraints. The high-level task parameterization defines the current task instance as it describes all varying factors that are important for successful task solving. In addition to a kinematic representation of a task, i.e. trajectories in joint angles or cartesian space, this thesis introduces primitive based generalization of forward signals that support the low-level controller in precise execution of motions. Those forward signals represent unmodeled dynamics and compensate for repetitive disturbances during task execution. This allows to perform high-level skill learning on complex robotic systems with unmodeled dynamic properties. The representation of dynamics in relation to a high-level task parameterization is not limited to the properties of the robot, dynamics of complex interactions can be represented as well. As a study case for complex robotic systems, this thesis refers to highly a compliant continuum trunk-shaped soft robot and a pneumatically driven humanoid child robot. For the acquisition of a skill, the parameterized skill consolidates parameterizations of successful actions for specific task instances. The required successful task instances can be gathered by kinesthetic teaching or by optimization with state-of-the-art reinforcement learning methods.

Further contributions of this thesis can be classified into two scopes, as discussed in the following:

Efficient Exploration of Parameterized Skills In case the parameterized skill is trained with solutions of an optimizer, the designed reward function is a crucial aspect for a good generalization performance of the parameterized skill. This thesis shows that additional cost terms can support consistent training samples without ambiguities caused by the redundancy of the task solutions, which are introduced as a regularization of the reward. To reduce the number of trials the optimizer has to perform to acquire a skill, this thesis proposes a bootstrapping mechanism. Previous experience is used to enhance the initial conditions for optimization of unsolved task instances. Evaluation of the aforementioned methods shows a significant reduction of the required trials as well as an improved generalization of the parameterized skill. Further, task related manifolds are investigated to achieve an enhancement of the efficiency of the skill learning. This thesis proposes a novel optimization scheme that performs a hybrid optimization in the task and the policy space. This allows a combination of a fast coarse optimization and slow fine tuning of actions. Evaluation shows the applicability of the hybrid optimization for robotic scenarios in simulation and on a real robotic setup. Additionally, a transfer learning approach for the parameterized skill is presented that allows a quick (in terms of trials) adaptation to altered perceptions.

Skill Learning on Highly Compliant Robotic Systems Real robotic systems with complex dynamic properties suffer from the lack of proper feed-forward control. Therefore, the execution of precise movements is limited and complex task learning is hardly feasible. Learning approaches that estimate an inverse model of the complete robotic system suffer from the huge state space. For this reason, this thesis explores low-level control of highly compliant actuators that is improved by learned inverse equilibrium models. The inverse equilibrium models capture simplified dynamic properties, i.e. only stable postures of the robot. This thesis demonstrates that classical feedback control in combination with learned inverse equilibrium models leads to an improved control on two pneumatically driven robotic platforms. Additionally, an interactive control mode is evaluated that provides kinesthetic teaching and human-robot-interaction on complex robotic platforms.

Finally, this thesis examines the feasibility of parameterized skills for generalization of the aforementioned feed-forward signals that support the feedback controller for control of highly compliant robots. Subsequently to an evaluation of the generalization capabilities of parameterized skills for forward signals, an integration of kinematic representations and the representation of dynamic properties is pursued. Demonstration is performed on a complex task that involves kinesthetic teaching, interaction with the environment, control of a 8-DOF pneumatically actuated robot, and parameterized task conditions.

7.1 Outlook

The promising results of this thesis motivate further research regarding high-level skill learning. In the following, some aspects will be elaborated:

Applicability to Further Robotic Systems The proposed framework for skill learning is not limited to pneumatically actuated robots and as demonstrated in simulation, not limited to an antagonistic actuation principle. As argued in this thesis, the generalization of task related forward signals is able to compensate for repetitive disturbances. As long as the repeatability of the robot is present, i.e. similar control signals result in similar movements, it can be assumed that the proposed control scheme is applicable. As an example, the control of tendon driven robots or robots driven by pneumatic muscles could benefit from the proposed skill learning framework.

Scalability to a Higher Complexity The experiments in this thesis demonstrated skill learning for robotic systems with up to 10-DOF. One of the humanoid robot systems with the highest complexity in terms of DOF is presented by Asano et al. [2017]. It incorporates over 100 tendon driven actuators. From the point of view of the generalization of the parameterized skill, learning is independent of the output dimensionality as the parameterized skill performs a task related generalization. Further, state-of-the-art optimization is able to handle high-dimensionalities of the optimization problem. However, further advanced concepts that aim at skill learning and learning of inverse models could be integrated in the skill learning framework, like goal babbling Rolf et al. [2010] or skill babbling Reinhart [2017], that efficiently deal with high dimensional learning problems.

The proposed skill learning methods assume a simplification of optimization of forward signals: optimization of forward signals is performed independently per joint. The applied method ILC, is very efficient as it follows a gradient information. But independent optimization is may not sufficient to find good solutions for robotic systems with an enhanced complexity. Therefore, a less efficient optimization based on a global reward (e.g. based on combined tracking error) could be beneficial. Due to the availability of a fast and not precise optimization method and an additional precise optimization that is costly, the application of the hybrid optimization (HCMA-ES), that is presented in this thesis, could be considered to combine the benefits of both methods.

An Enriched Representation of Skills A further interesting issue is the extension of the proposed skill learning architecture for mixtures of primitives and sequencing of primitives. Mixtures of primitives, for example, is a common technique to generate new motion patterns from a previously learned task set. As demonstrated in this thesis, the parameterized skill is able to successfully generalize forward signals for a changing task parameterization, this raises the question of how inter-primitive generalization performs. In particular with consideration of primitives that incorporate generalized forward signals of multiple parameterized skills.

Appendix A

Appendix

A.1 Parameter Grid Seach for Inverse Equilibrium Model of the Affetto Robot

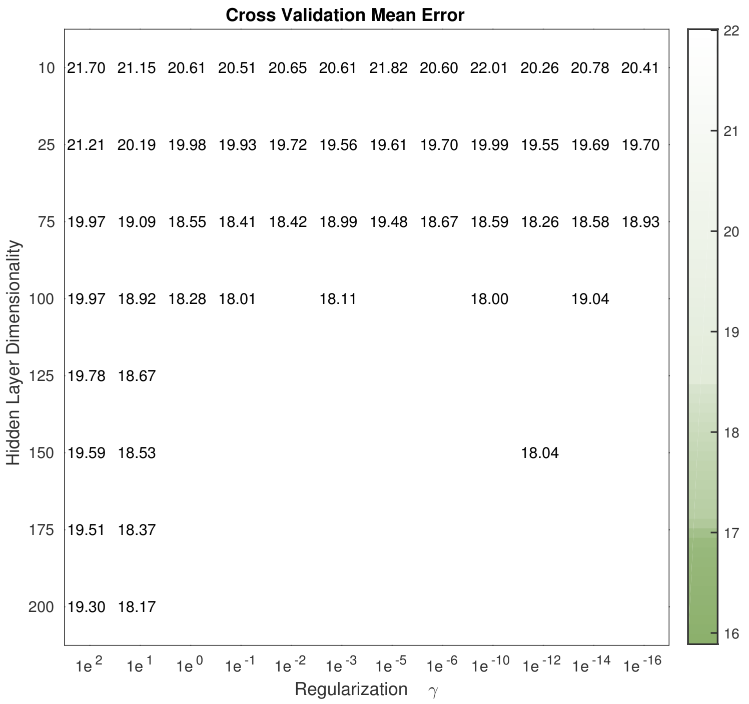


Figure 1.1: Cross-validation error for learning of the inverse equilibrium model. $R = 125$ and $\lambda = 1$ have been selected for a compromise between a low error and a low deviation of the solutions.

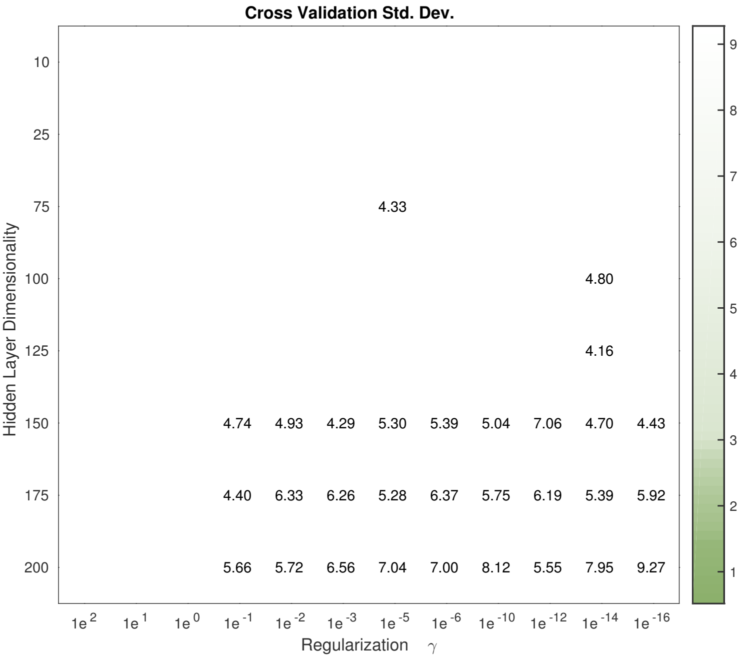
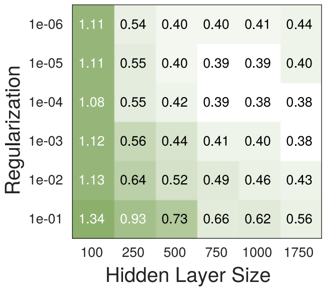


Figure 1.2: Standard deviation for learning of the inverse equilibrium model. $R = 125$ and $\lambda = 1$ have been selected for a compromise between a low error and a low deviation of the solutions.

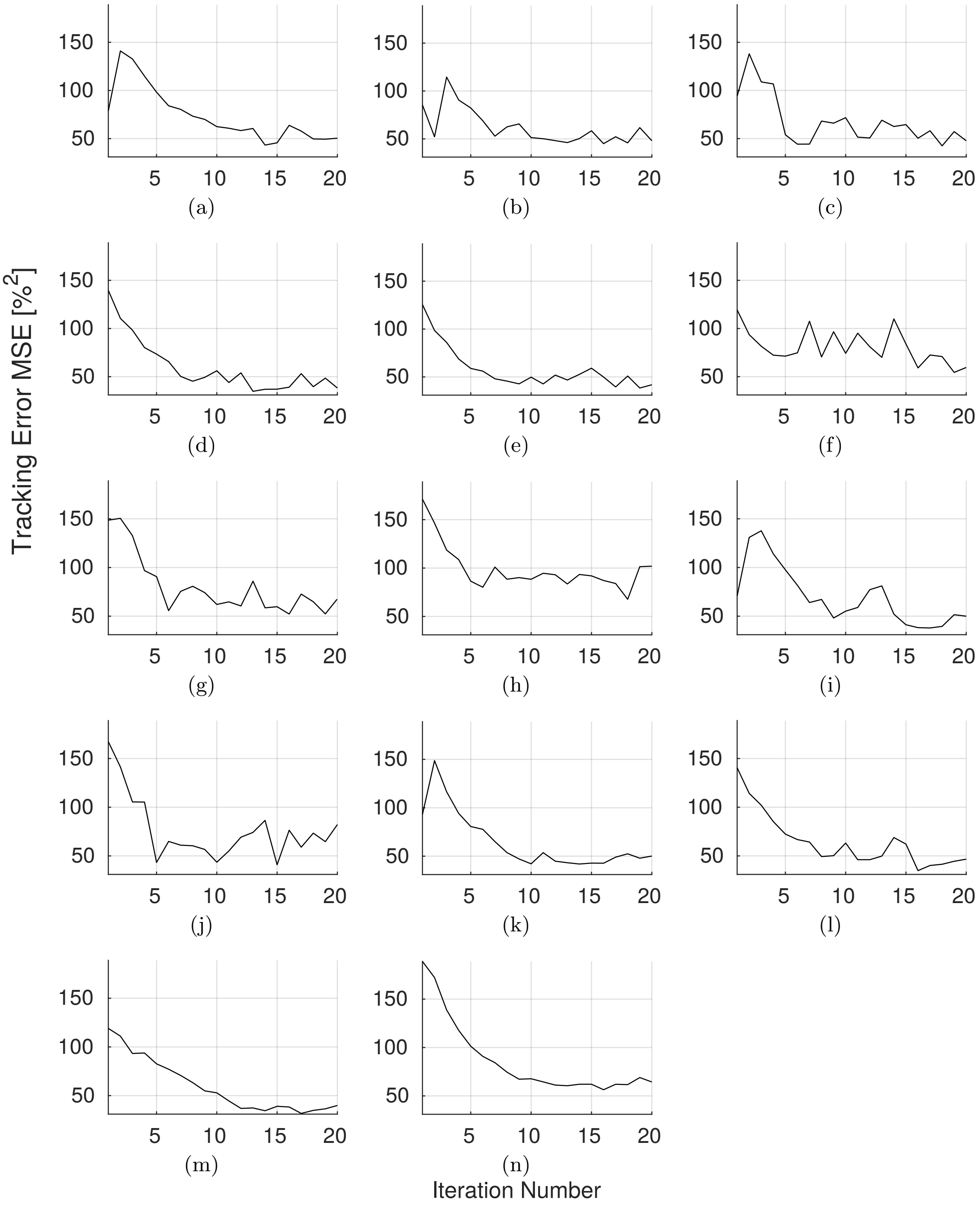
A.2 Parameter Grid Seach for Inverse Equilibrium Model of the UR5 Robot



Regularization \ Hidden Layer Size	100	250	500	750	1000	1750
1e-06	1.11	0.54	0.40	0.40	0.41	0.44
1e-05	1.11	0.55	0.40	0.39	0.39	0.40
1e-04	1.08	0.55	0.42	0.39	0.38	0.38
1e-03	1.12	0.56	0.44	0.41	0.40	0.38
1e-02	1.13	0.64	0.52	0.49	0.46	0.43
1e-01	1.34	0.93	0.73	0.66	0.62	0.56

Figure 1.3: Cross-validation error for learning the inverse equilibrium model. Parameterization $R = 500$ hidden neurons and a regularization of $\gamma = 10^{-5}$ were selected for learning of the inverse equilibrium model.

A.3 Optimization of Human Demonstrations



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n)

Figure 1.4: Tracking error during optimization of forward signals by ILC for demonstrated movements. All movements solve the task after optimization ($R \geq 0.85$).

A.4 Example Task Instances of the Drumming Scenario

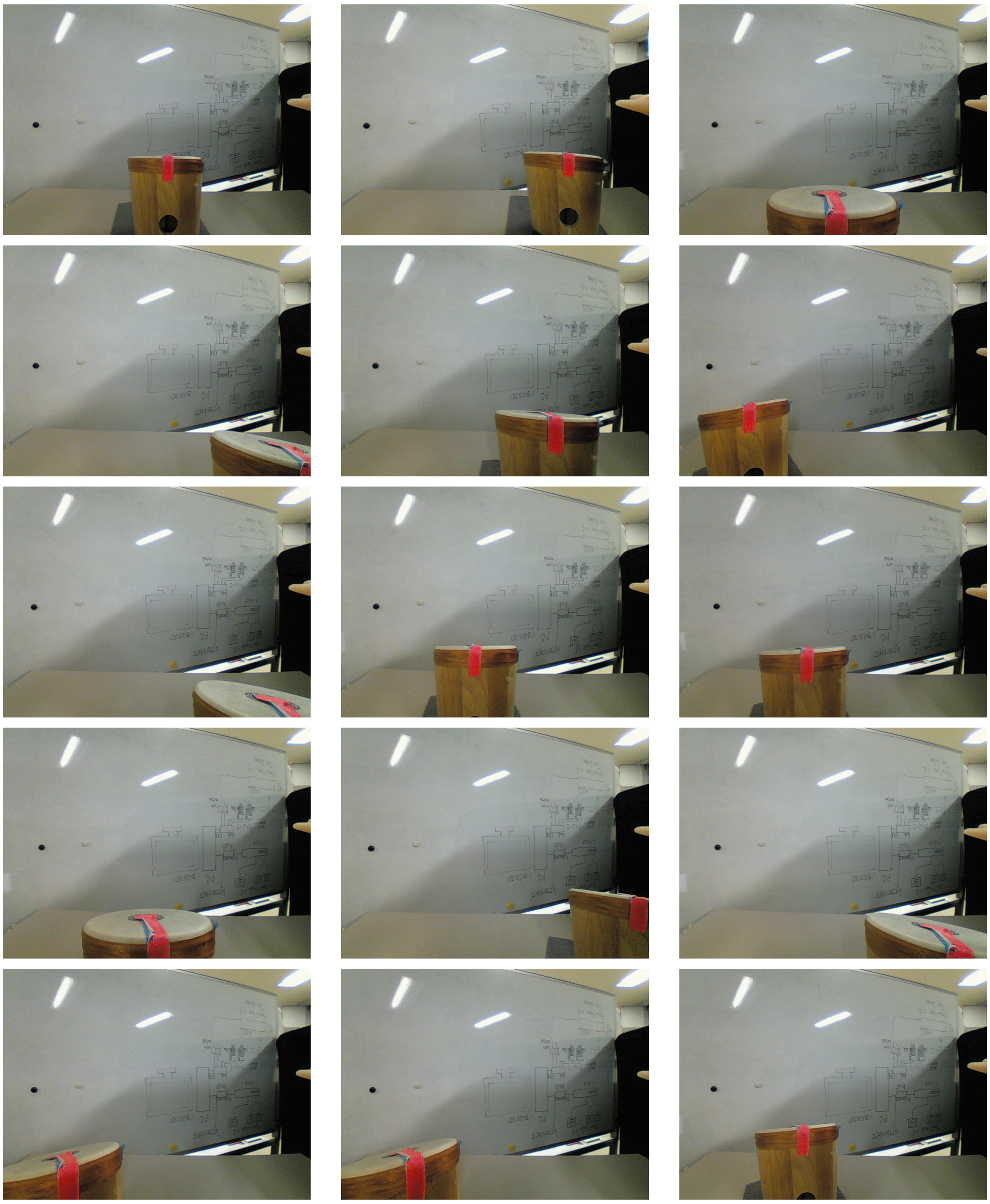


Figure 1.5: Examples of randomly selected positions in the workspace of the Affetto drum Scenario.

A.5 Prototype Spectra of Human Demonstrations

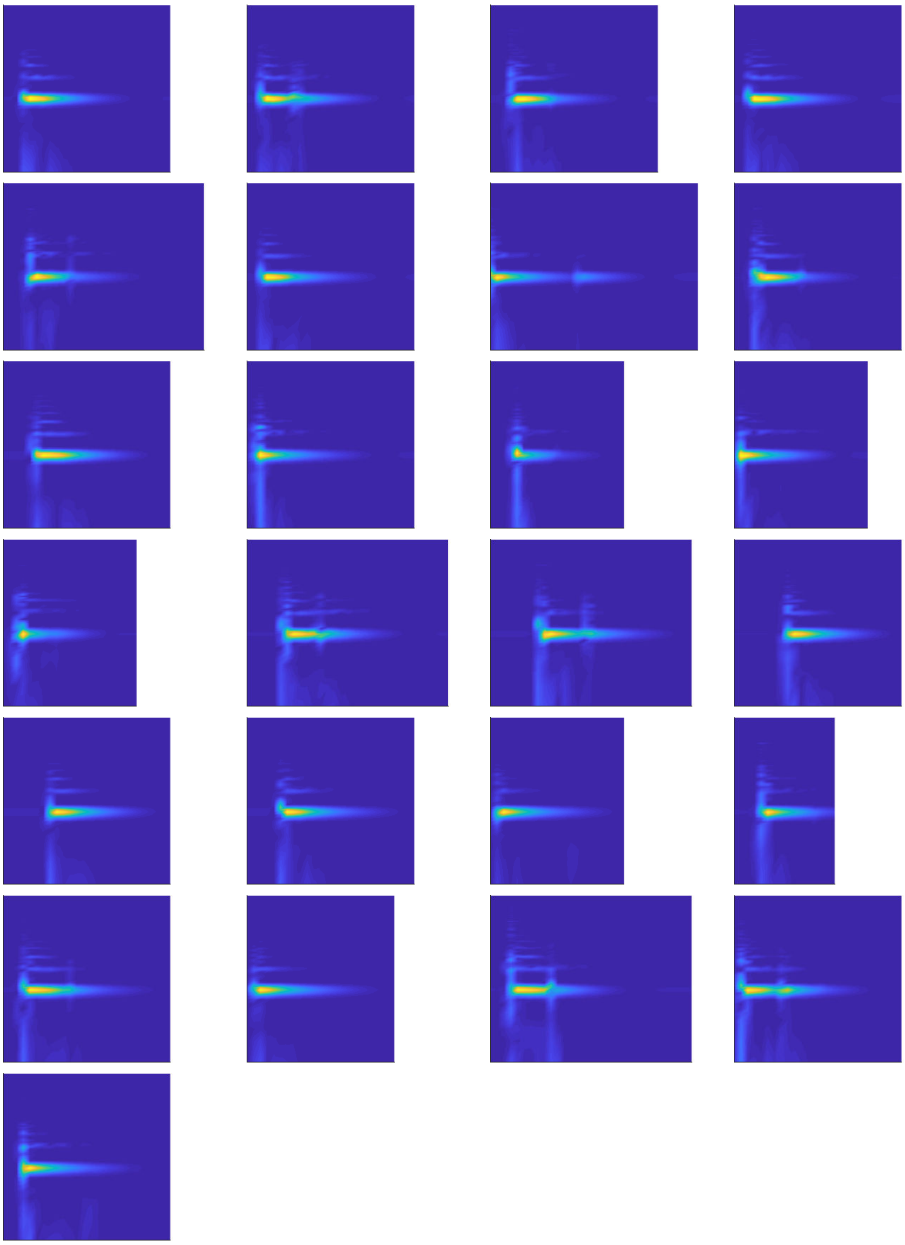


Figure 1.6: Spectrograms of positive prototypes of drumming actions. Actions are recorded by kinesthetic teaching and executed on the robot.

A.6 Interactive Scenario: Joint Angle Trajectories

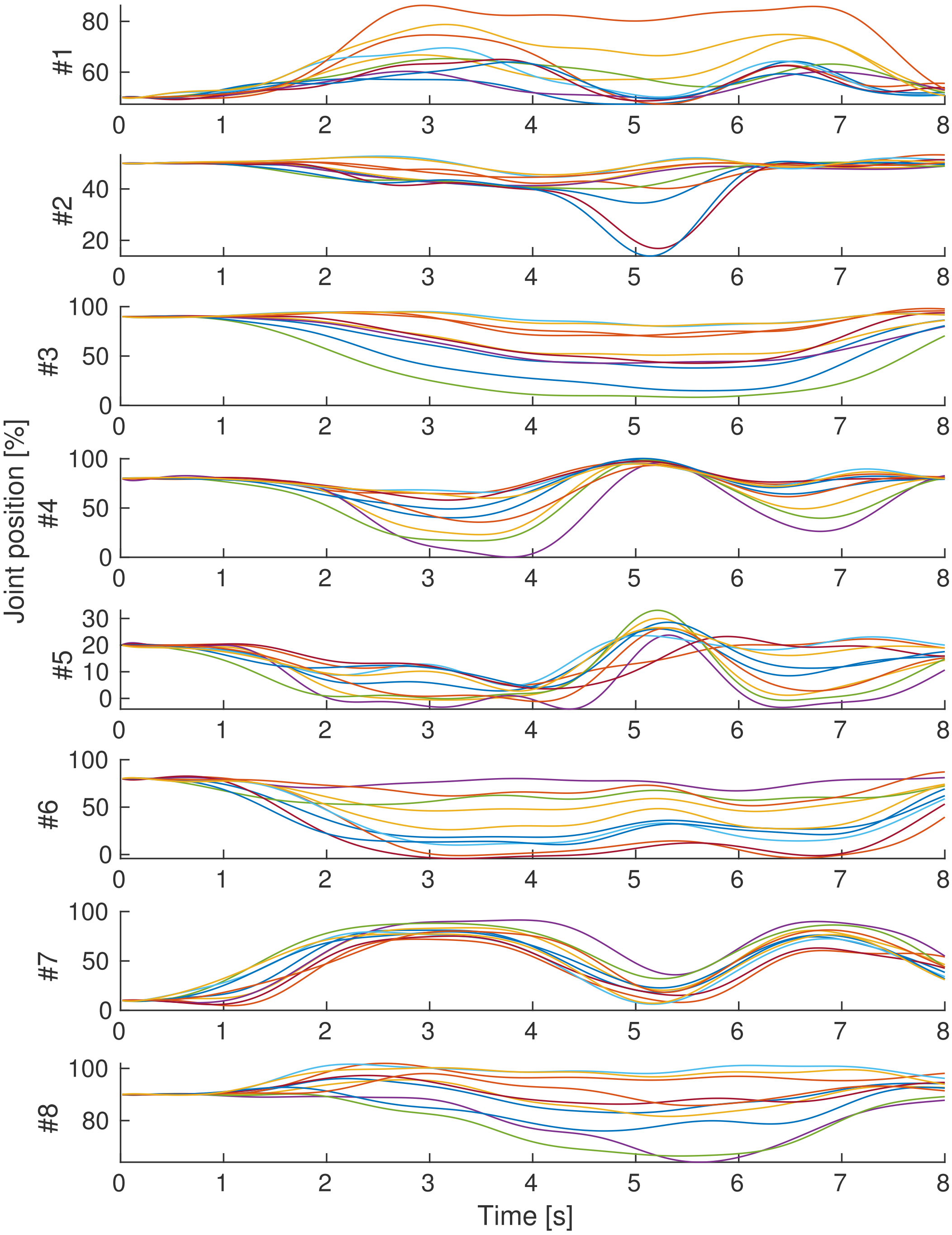


Figure 1.7: Generalized joint angle trajectories of the interaction scenario. Results for all ten task parameterizations of the evaluation, mean of 10 repetitions.

A.7 Interactive Scenario: Optimized Forward Signals

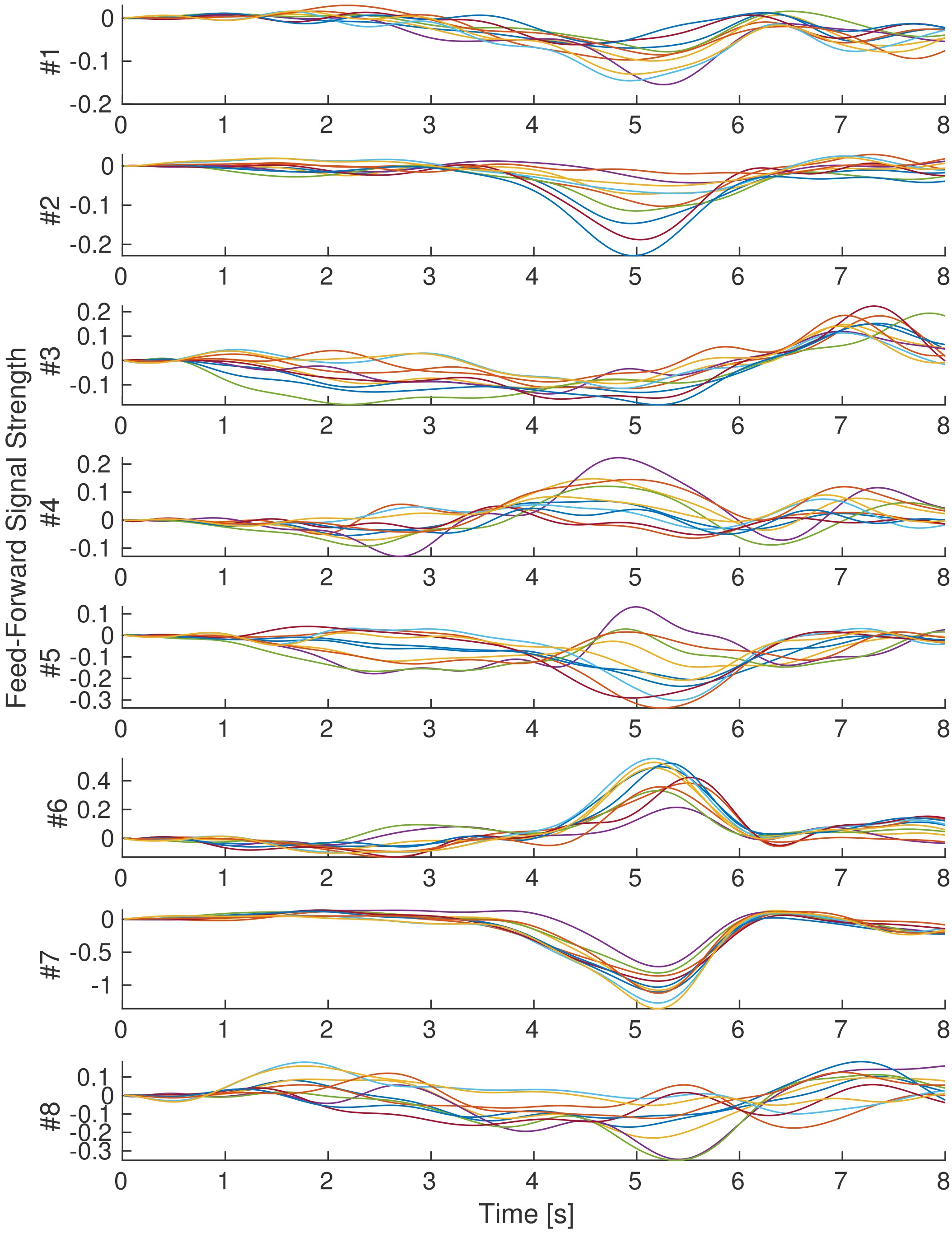
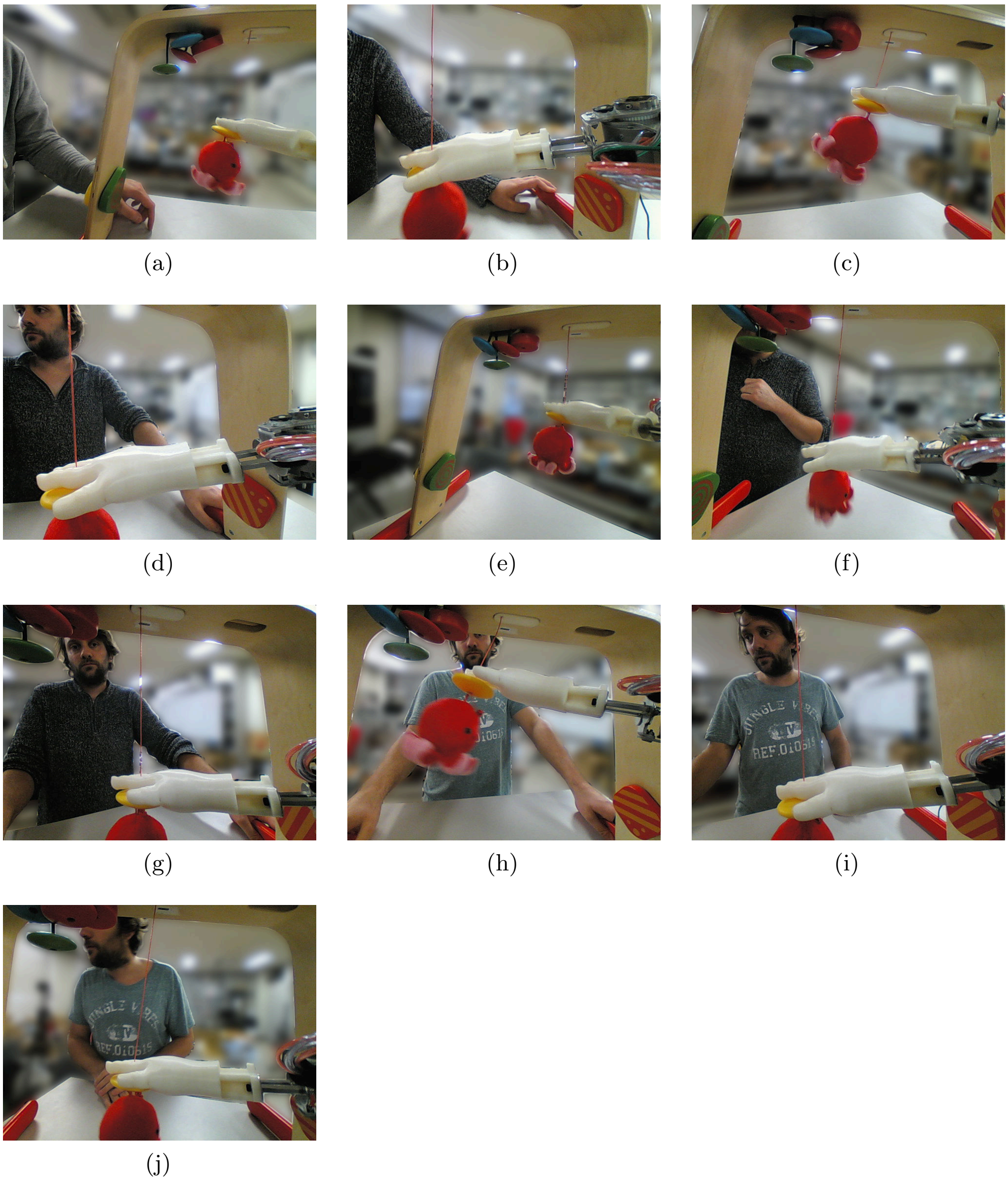


Figure 1.8: Generalized forward signals of the interaction scenario. Results for all ten task parameterizations of the evaluation, mean of 10 repetitions.

A.8 Interactive Scenario: Sucessful Generalizations



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 1.9: Snapshots of successful actions that are generalized by the parameterized skill. Joint angle trajectories and forward signals are used for motion execution.

References

Adams, J. A.
1971. A closed-loop theory of motor learning. *Journal of Motor Behavior*, 3(2):111–150. Cited on page 15.

Adams, J. A.
1987. Historical review and appraisal of research on the learning, retention, and transfer of human motor skills. *Psychological bulletin*, 101(1):41. Cited on page 15.

Adolph, K. and S. R. Robinson
2013. *The Road to Walking: What Learning to Walk Tells Us About Development*, Pp. 403–443. Oxford University Press. Cited on page 3.

Aggarwal, C. C., A. Hinneburg, and D. A. Keim
2001. On the surprising behavior of distance metrics in high dimensional space. In *Database Theory — ICDT 2001*, J. Van den Bussche and V. Vianu, eds., Pp. 420–434, Berlin, Heidelberg. Springer Berlin Heidelberg. Cited on page 27.

Akgun, B., M. Cakmak, J. W. Yoo, and T. A. Lockerd
2012. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, Pp. 391–398, New York, NY, USA. ACM. Cited on page 18.

Aksoy, E. E., A. Orhan, and F. Wörgötter
2016. Semantic decomposition and recognition of long and complex manipulation action sequences. *International journal of computer vision*, 122(1):84–115. Cited on page 3.

Aksoy, E. E., M. Tamosiunaite, and F. Wörgötter
2015. Model-free incremental learning of the semantics of manipulation actions.

Robotics and Autonomous Systems, 71:118–133. Emerging Spatial Competences: From Machine Perception to Sensorimotor Intelligence. Cited on page 3.

Albu-Schäffer, A., O. Eiberger, M. Fuchs, M. Grebenstein, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, S. Wolf, C. Borst, and G. Hirzinger
2011. Anthropomorphic soft robotics – from torque control to variable intrinsic compliance. In *Robotics Research*, Pp. 185–207. Springer Berlin Heidelberg. Cited on page 16.

Albu-Schäffer, A., S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger
2007. The dlr lightweight robot: Design and control concepts for robots in human environments. *Industrial Robot: the International journal of robotics research and application*, 34(5):376–385. Cited on pages 2 and 16.

An, Q., Y. Ishikawa, T. Funato, S. Aoi, H. Oka, H. Yamakawa, A. Yamashita, and H. Asama
2014. Generation of human standing-up motion with muscle synergies using forward dynamic simulation. In *IEEE International Conference on robotics and automation (ICRA)*, Pp. 730–735. IEEE. Cited on page 18.

Andersson, R. L.
1989. Aggressive trajectory generator for a robot ping-pong player. *IEEE Control Systems Magazine*, 9(2):15–21. Cited on page 31.

Arimoto, S., S. Kawamura, and F. Miyazaki
1984. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140. Cited on pages 35 and 131.

Asano, Y., K. Okada, and M. Inaba
2017. Design principles of a human mimetic humanoid: Humanoid platform to study human intelligence and internal body system. *Science Robotics*, 2(13). Cited on page 151.

Balayn, A., J. F. Queißer, M. Wojtynek, and S. Wrede
2016. Adaptive handling assistance for industrial lightweight robots in simulation. In *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, Pp. 1–8. Cited on page 122.

Baranes, A. and P. Oudeyer
2013. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73. Cited on pages 4, 19, 40, 126, and 127.

Ben-David, S., J. Blitzer, K. Crammer, and F. Pereira
2006. Analysis of representations for domain adaptation. In *Proceedings of the 19th Advances in Neural Information Processing Systems Conference (NIPS 2006)*, Pp. 137–144. Cited on page 87.

Bernstein, N. A.
1967. *The Co-Ordination and Regulation of Movements*. New York: Pergamon Press. Cited on page 14.

Bishop, C. M.
2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. Cited on page 27.

Blöbaum, P., A. Schulz, and B. Hammer
2015. Unsupervised dimensionality reduction for transfer learning. In *Proceedings. 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Pp. 507–512. Cited on page 87.

Bristow, D. A., M. Tharayil, and A. G. Alleyne
2006. Survey of iterative learning control: A learning-based method for high-performance tracking control. *IEEE Control Systems*, 26(3):96–114. Cited on pages vii, 36, and 131.

Bryan, W. L. and N. Harter
1897. Studies in the physiology and psychology of the telegraphic language. *Psychological Review*, 4(1):27. Cited on page 14.

Burke, R. E.
2007. Sir charles sherrington's the integrative action of the nervous system: a centenary appreciation. *Brain : a journal of neurology*, 130(4):887–894. Cited on page 14.

Buss, S. R.
2004. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Technical report, IEEE Journal of Robotics and Automation. Cited on page 74.

Büchler, D., H. Ott, and J. Peters
2016. A lightweight robotic arm with pneumatic muscles for robot learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Pp. 4086–4092. Cited on page 17.

Cai, C. and H. Jiang
2013. Performance comparisons of evolutionary algorithms for walking gait optimization. In *IEEE International Conference on Information Science and Cloud Computing Companion (ISCC-C)*, Pp. 129–134. IEEE. Cited on page 19.

Calinon, S.
2016. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics*, 9(1):1–29. Cited on page 20.

Calinon, S., T. Alizadeh, and D. G. Caldwell
2013. On improving the extrapolation capability of task-parameterized movement models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Pp. 610–616. Cited on page 20.

Calinon, S., Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell
2012. Statistical dynamical systems for skills acquisition in humanoids. In *12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, Pp. 323–329, Osaka, Japan. Cited on page 31.

Calisti, M., M. Giorelli, G. G. Levy, B. Mazzolai, B. Hochner, C. Laschi, and P. Dario
2011. An octopus-bioinspired solution to movement and manipulation for soft robots. *Bioinspiration & Biomimetics*, 6(3):036002. Cited on page 17.

Chartrand, R. and W. Yin
2008. Iteratively reweighted algorithms for compressive sensing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Pp. 3869–3872. Cited on page 27.

Chen, C.-K. and J. Hwang
2005. Iterative learning control for position tracking of a pneumatic actuated x-y table. *Control Engineering Practice*, 13(12):1455–1461. Cited on pages 36 and 131.

Cho, K. H., T. Raiko, and A. Ilin
2013. Gaussian-bernoulli deep boltzmann machine. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Pp. 1–7. Cited on page 29.

Colasanto, L., N. G. Tsagarakis, and D. G. Caldwell
2012. A compact model for the compliant humanoid robot coman. In *4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Pp. 688–694. Cited on page 53.

Colome, A., G. Neumann, J. Peters, and C. Torras
2014. Dimensionality reduction for probabilistic movement primitives. In *14th IEEE-RAS International Conference on Humanoid Robots, Humanoids*, Pp. 794–800. Cited on page 66.

Daniel, C., O. Kroemer, M. Viering, J. Metz, and J. Peters
2015. Active reward learning with a novel acquisition function. *Autonomous Robots*, 39(3):389–405. Cited on page 45.

Dehio, N., R. F. Reinhart, and J. J. Steil
2016. Continuous task-priority rearrangement during motion execution with a mixture of torque controllers. In *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Pp. 264–270. Cited on page 46.

Deng, L. and D. Yu
2014. Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387. Cited on page 24.

Deng, W., Q. Zheng, and L. Chen
2009. Regularized extreme learning machine. *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, Pp. 389–395. Cited on page 27.

Devol Jr., G. C.
1954. Programmed article transfer. patent us 2988237 a. US. Cited on page 16.

Ding, L., H. Wu, Y. Yao, and Y. Yang
2015. Dynamic model identification for 6-dof industrial robots. *Journal of Robotics*, 2015:9. Cited on page 120.

Duchaine, V., N. Lauzier, M. Baril, M. A. Lacasse, and C. Gosselin
2009. A flexible robot skin for safe physical human robot interaction. In *IEEE International Conference on Robotics and Automation (ICRA)*, Pp. 3676–3681. Cited on page 17.

Edwards, W. H.
2010. *Motor Learning and Control: From Theory to Practice*. Cengage Learning. Cited on pages 14 and 15.

Emmerich, C., R. F. Reinhart, and J. J. Steil
2013. Multi-directional continuous association with input-driven neural dynamics. *Neurocomputing*, 112:47–57. Advances in artificial neural networks, machine learning, and computational intelligence. Cited on pages 4 and 30.

Enache, M. and R. Dogaru
2015. A benchmark study regarding extreme learning machine, modified versions of naïve bayes classifier and fast support vector classifier. In *E-Health and Bioengineering Conference (EHB)*, Pp. 1–4. Cited on page 25.

Endres, F., J. Trinkle, and W. Burgard
2013. Learning the dynamics of doors for robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Pp. 3543–3549. Cited on page 1.

Fabisch, A., Y. Kassahun, H. Wöhrle, and F. Kirchner
2013. Learning in compressed space. *Neural Networks*, 42:83–93. Cited on page 65.

Fachantidis, A., I. Partalas, M. E. Taylorand, and IoannisVlahavas
2012. Transfer learning via multiple inter-task mappings. In *Recent Advances in Reinforcement Learning*, S. Sanner and M. Hutter, eds., Pp. 225–236, Berlin, Heidelberg. Springer Berlin Heidelberg. Cited on page 87.

Flash, T. and N. Hogan
1984. The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7):1688–1703. Cited on page 51.

Fligge, N., J. McIntyre, and P. van der Smagt
2012. Minimum jerk for human catching movements in 3D. In *4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Pp. 581–586. Cited on page 51.

Franzius, M., H. Sprekeler, and L. Wiskott
2007. Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLOS Computational Biology*, 3(8):1–18. Cited on page 24.

Freund, Y. and D. Haussler
1992. Unsupervised learning of distributions on binary vectors using two layer networks. In *Advances in Neural Information Processing Systems 4*, Pp. 912–919. Cited on page 29.

Ghahramani, Z. and M. I. Jordan
1994. Supervised learning from incomplete data via an em approach. In *Advances in neural information processing systems 6*, J. D. Cowan, G. Tesauro, and J. Alspector, eds., Pp. 120–127. Morgan-Kaufmann. Cited on page 33.

Girosi, F., M. Jones, and T. Poggio
1995. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269. Cited on page 46.

Glaubius, R. and W. D.Smart
2005. Manifold representations for continuous-state reinforcement learning. Technical report, Department of Computer Science and Engineering, Washington University. Cited on page 65.

Glaubius, R., M. Namihira, and W. D. Smart
2005. Speeding up reinforcement learning using manifold representations: Preliminary results. In *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR)*, Pp. 62–70. Cited on page 65.

Goto, S., N. Nakamura, and N. Kyura
2003. Forcefree control with independent compensation for inertia friction and gravity of industrial articulated robot arm. In *IEEE International Conference on Robotics and Automation*, volume 3, Pp. 4386–4391. Cited on page 120.

Graziano, M. S. A.
2015a. *Brain Mapping: An Encyclopedic Reference*, chapter Cortical Action Representations, Pp. 683–686. Elsevier Science & Technology. Cited on page 19.

Graziano, M. S. A.
2015b. *Shared Representations: Sensorimotor Foundations of Social Life*, chapter A new view of the motor cortex. UK: Cambridge University Press. Cited on pages 18 and 127.

Grzesiak, A., R. Becker, and A. Verl
2011. The bionic handling assistant - a success story of additive manufacturing. *Assembly Automation*, 31(4):329–333. Cited on pages 17 and 96.

Guizzo, E. and E. Ackerman
2015. The hard lessons of darpa's robotics challenge [news]. *IEEE Spectrum*, 52(8):11–13. Cited on page 1.

Günter, F.
2009. *Using reinforcement learning for optimizing the reproduction of tasks in robot programming by demonstration*. PhD thesis, STI, Lausanne. Cited on page 19.

Günter, F., M. Hersch, S. Calinon, and A. Billard
2007. Reinforcement learning for imitating constrained reaching movements. *Advanced Robotics, Special Issue on Imitative Robots*, 21(13):1521–1544. Cited on pages 32, 33, 43, 68, and 130.

Ham, R. V., T. G. Sugar, B. Vanderborght, K. W. Hollander, and D. Lefeber
2009. Compliant actuator designs. *IEEE Robotics Automation Magazine*, 16(3):81–94. Cited on page 16.

Hansen, N.
2006. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, eds., Pp. 75–102. Springer. Cited on pages 34, 35, 41, 44, 68, 70, and 71.

Haq, A., Y. Aoustin, and C. Chevallereau
2011. Compliant joints increase the energy efficiency of bipedal robot. In *The 14th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR2011)*, Paris, France. 6. Cited on page 3.

Hauser, H., A. J. Ijspeer, R. M. Füchslin, R. Pfeifer, and W. Maass
2011. Towards a theoretical foundation for morphological computation with compliant bodies. *Biological Cybernetics*, 105(5-6):355–370. Cited on page 17.

Helwa, M. K. and A. P. Schoellig
2017. Multi-robot transfer learning: A dynamical system perspective. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Pp. 4702–4708. Cited on page 87.

Hillerström, G. and K. Walgama
1996. Repetitive control theory and applications - a survey. *IFAC Proceedings Volumes*, 29(1):1446–1451. 13th World Congress of IFAC, 1996, San Francisco USA, 30 June - 5 July. Cited on page 34.

Hinton, G. E.
2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800. Cited on page 30.

Hinton, G. E. and R. R. Salakhutdinov
2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507. Cited on page 29.

Hirzinger, G., N. Sporer, A. Albu-Schaffer, M. Hahnle, R. Krenn, A. Pascucci, and M. Schedl
2002. Dlr's torque-controlled light weight robot iii - are we reaching the technological limits now? In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 2, Pp. 1710–1716. Cited on pages 2 and 16.

Hopfield, J. J.
1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558. Cited on page 28.

Hopfield, J. J.
1984. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 81(10):3088–3092. Cited on page 28.

Hoshino, K.
2008. Control of speed and power in a humanoid robot arm using pneumatic actuators for human-robot coexisting environment. *IEICE Transactions on Information and Systems*, E91.D(6):1693–1699. Cited on page 98.

Hosoda, K., S. Sekimoto, Y. Nishigori, S. Takamuku, and S. Ikemoto
2012. Anthropomorphic muscular–skeletal robotic upper limb for understanding embodied intelligence. *Advanced Robotics*, 26(7):729–744. Cited on page 3.

Huang, G.-B., Q.-Y. Zhu, and C.-K. Siew
2006. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501. Neural NetworksSelected Papers from the 7th Brazilian Symposium on Neural Networks (SBRN '04)7th Brazilian Symposium on Neural Networks. Cited on pages 44, 68, and 131.

Huang, J., A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola
2007. Correcting sample selection bias by unlabeled data. In *Advances in Neural*

Information Processing Systems 19, B. Schölkopf, J. C. Platt, and T. Hoffman, eds., Pp. 601–608. MIT Press. Cited on page 87.

Hull, C. L.
1952. *A Behavior System; An Introduction to Behavior Theory Concerning the Individual Organism*. Yale University Press. Cited on page 15.

Huynh, H. T. and Y. Won
2009. Online training for single hidden-layer feedforward neural networks using RLS-ELM. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Pp. 469–473. Cited on page 44.

Huynh, H. T. and Y. Won
2011. Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks. *Pattern Recognition Letters*, 32(14):1930–1935. Cited on pages 27 and 49.

Hwang, J.-H., R. C. Arkin, and D.-S. Kwon
2003. Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 2, Pp. 1444–1449. Cited on page 31.

Ijspeert, A. J., A. Crespi, J.-M. Cabelguen, A. Ijspeert, A. Crespi, and J.-M. Cabelguen
2005. Simulation and robotics studies of salamander locomotion: Applying neurobiological principles to the control of locomotion in robots. *Neuroinformatics*, 3(3):171–195. Cited on page 17.

Ijspeert, A. J., J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal
2013. Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Computation*, 25(2):328–373. Cited on pages 4, 32, 43, 68, 125, and 130.

Ijspeert, A. J., J. Nakanishi, and S. Schaal
2002. Learning attractor landscapes for learning motor primitives. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, Pp. 1547–1554, Cambridge, MA, USA. MIT Press. Cited on page 84.

Ikemoto, S., Y. Kimoto, and K. Hosoda
2015. Shoulder complex linkage mechanism for humanlike musculoskeletal robot arms. *Bioinspiration & biomimetics*, 10(6):066009. Cited on page 17.

Ishihara, H.
2016. Compliant and compact joint mechanism for a child android robot. In *11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Pp. 553–554. Cited on page 97.

Ishihara, H. and M. Asada
2015. Design of 22-dof pneumatically actuated upper body for child android 'affetto'. *Advanced Robotics*, 29(18):1151–1163. Cited on pages xii, xiv, 97, 98, 126, and 135.

Ishihara, H., Y. Yoshikawa, and M. Asada
2011. Realistic child robot *Affetto* for understanding the caregiver-child attachment relationship that guides the child development. In *2011 IEEE International Conference on Development and Learning (ICDL)*, volume 2, Pp. 1–5. Cited on pages xii, xiv, 97, 98, 126, 127, and 135.

Jain, A., H. Nguyen, M. Rath, J. Okerman, and C. C. Kemp
2010. The complex structure of simple devices: A survey of trajectories and forces that open doors and drawers. In *3rd IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, Pp. 184–190. Cited on page 1.

James, W.
1890. *The Principles of Psychology*. New York: Holt and company. Cited on page 14.

Jefferys, W. H. and J. O. Berger
1992. Ockham's razor and bayesian analysis. *American Scientist*, 80(1):64–72. Cited on page 46.

Kawai, R., T. Markman, R. Poddar, R. Ko, A. L. Fantana, A. K. Dhawale, A. R. Kampff, and B. P. Ölveczky
2015. Motor cortex is required for learning but not for executing a motor skill. *Neuron*, 86(3):800–812. Cited on page 18.

Kawai, Y., J. Park, T. Horii, Y. Oshima, K. Tanaka, H. Mori, Y. Nagai, T. Takuma, and M. Asada
2012. Throwing skill optimization through synchronization and desynchronization of degree of freedom. In *16th Annual RoboCup International Symposium*, Pp. 178–189. Cited on page 66.

Kawato, M., K. Furukawa, and R. Suzuki
1987. A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, 57(3):169–185. Cited on page 127.

Kawato, M., Y. Uno, M. Isobe, and R. Suzuki
1988. Hierarchical neural network model for voluntary movement with application to robotics. *IEEE Control Systems Magazine*, 8(2):8–15. Cited on page 126.

Kim, D.-I. and S. Kim
1996. An iterative learning control method with application for cnc machine tools. *IEEE Transactions on Industry Applications*, 32(1):66–72. Cited on pages 36 and 131.

Kober, J. and J. Peters
2010. Policy search for motor primitives in robotics. *Machine Learning*, 84(1):171–203. Cited on pages 4 and 125.

Kober, J. and J. Peters
2012. *Reinforcement Learning in Robotics: A Survey*, volume 12, Pp. 579–610. Berlin, Germany: Springer. Cited on page 45.

Kober, J., A. Wilhelm, E. Oztop, and J. Peters
2012. Reinforcement learning to adjust parametrized motor primitives to new situations. *Autonomous Robots*, 33:361–379. 10.1007/s10514-012-9290-3. Cited on pages 4, 19, 23, 40, and 126.

Korane, K. J.
2010. Robot imitates nature. *Machine Design*, 82(18):68–70. Cited on page 96.

Koutnik, J., F. Gomez, and J. Schmidhuber
2010. Evolving neural networks in compressed weight space. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, Pp. 619–626, New York, NY, USA. ACM. Cited on page 65.

Krahe, N.
1999. Motor skill learning: A review of trends and theories. Master's thesis, Undergraduate Theses. Carroll College, Helena, MT, Helena, MT. Cited on page 15.

Kulvicius, T., K. Ning, M. Tamosiunaite, and F. Wörgötter
2012. Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting. *IEEE Transactions Robotics*, 28(1):145–157. Cited on pages 32 and 44.

Lattal, K. A.
1998. A century of effect: Legacies of el thorndike's animal intelligence monograph. *Journal of the experimental analysis of behavior*, 70(3):325–336. Cited on page 14.

LeCun, Y., Y. Bengio, and G. E. Hinton
2015. Deep learning. *Nature*, 521(7553):436–444. Insight. Cited on page 24.

Lemme, A., A. Freire, G. Barreto, and J. J. Steil
2013. Kinesthetic teaching of visuomotor coordination for pointing by the humanoid robot icub. *Neurocomputing*, 112:179–188. Cited on page 18.

Lemme, A., K. Neumann, R. F. Reinhart, and J. J. Steil
2014. Neural learning of vector fields for encoding stable dynamical systems. *Neurocomputing*, 141(0):3–14. Cited on pages 33, 43, 68, and 130.

Liang, N., G. Huang, P. Saratchandran, and N. Sundararajan
2006. A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on Neural Networks*, 17(6):1411–1423. Cited on pages 26 and 44.

Liegeois, A.
1977. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions Systems, Man and Cybernetics*, 7(12):842–868. Cited on page 54.

Liu, G.-R. and X. Han
2003. *Computational Inverse Techniques in Nondestructive Evaluation*, chapter 4.6.2, Pp. 103–105. CRC Press. Cited on page 74.

Liu, X., C. Gao, and P. Li
2012. A comparative analysis of support vector machines and extreme learning machines. *Neural Networks*, 33:58 – 66. Cited on page 25.

Longman, R. W.
1998. *Designing Iterative Learning and Repetitive Controllers*, Pp. 107–146. Boston, MA: Springer US. Cited on pages 35 and 131.

Lungarella, M. and L. Berthouze
2002. On the interplay between morphological, neural, and environmental dynamics: A robotic case study. *Adaptive Behavior*, 10(3-4):223–241. Cited on page 66.

Malekzadeh, M., J. Queißer, and J. J. Steil
2017a. Imitation learning for a continuum trunk robot. In *Proceedings of the 25. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. (ESANN)*, M. Verleysen, ed. Ciaco. Cited on page 108.

Malekzadeh, M. S., S. Calinon, D. Bruno, and D. G. Caldwell
2014a. Learning by imitation with the STIFF-FLOP surgical robot: A biomimetic approach inspired by octopus movements. *Special Issue on Medical Robotics and Biomimetics*, 1(13):1–15. Cited on page 32.

Malekzadeh, M. S., S. Calinon, D. Bruno, and D. G. Caldwell
2014b. A skill transfer approach for continuum robots-imitation of octopus reaching motion with the stiff-flop robot. In *AAAI Symposium on Knowledge, Skill, and Behavior Transfer in Autonomous Robots*, Pp. 49–52. Cited on page 87.

Malekzadeh, M. S., J. F. Queißer, and J. J. Steil
[Submitted]. Control of bionic handling assistant robot by learning from demonstration. *Advanced Robotics*. Cited on pages 107 and 108.

Malekzadeh, M. S., J. F. Queißer, and J. J. Steil
2015. Learning from demonstration for bionic handling assistant robot. In *IROS 2015 Workshop - New Frontiers and Applications for Soft Robotics*, Pp. 101–107.

Malekzadeh, M. S., J. F. Queißer, and J. J. Steil.
2016. Learning the end-effector pose from demonstration for the bionic handling assistant robot. In *9th International Workshop on Human-Friedly Robotics*.

Malekzadeh, M. S., J. F. Queißer, and J. J. Steil
2017b. Imitation learning for a continuum trunk robot. In *Proceedings of the 25. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. ESANN 2017*, M. Verleysen, ed., Pp. 335–340. Ciaco.

Marques, H. G., M. Jäntsch, S. Wittmeier, O. Holland, C. Alessandro, A. Diamond, M. Lungarella, and R. Knight
2010. Ecce1: The first of a series of anthropomimetic musculoskeletal upper torsos. In *10th IEEE-RAS International Conference on Humanoid Robots*, Pp. 391–396. Cited on page 17.

Matsubara, T., S.-H. Hyon, and J. Morimoto
2011. Learning parametric dynamic movement primitives from multiple demonstrations. *Neural networks : the official journal of the International Neural Network Society*, 24(5):493–500. Cited on page 19.

McGeer, T.
1990. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82. Cited on page 3.

Moro, F., N. Tsagarakis, and D. G. Caldwell
2012. On the kinematic motion primitives (kmps) - theory and application. *Frontiers in Neurorobotics*, 6:10. Cited on page 66.

Mulder, T. and W. Hulstyn
1984. Sensory feedback therapy and theoretical knowledge of motor control and learning. *American journal of physical medicine*, 63(5):226–244. Cited on page 15.

Müller, V. C. and M. Hoffmann
2017. What is morphological computation? on how the body contributes to cognition and control. *Artificial Life*, 23(1):1–24. Cited on page 3.

Mülling, K., J. Kober, O. Kroemer, and J. Peters
2013. Learning to select and generalize striking movements in robot table tennis. *International Journal of Robotics Research*, 32(3):263–279. Cited on pages 4 and 126.

Mülling, K., J. Kober, and J. Peters
2010. Learning table tennis with a mixture of motor primitives. *Proceedings of the 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2010)*, Pp. 411–416. Cited on pages 19, 23, and 40.

Mussa-Ivaldi, F. A. and E. Bizzi
2000. Motor learning through the combination of primitives. In *Philosophical Transactions of the Royal Society B: Biological Sciences*, volume 2, Pp. 1–5. Cited on pages 3 and 18.

Narioka, K., S. Moriyama, and K. Hosoda
2011. 2p2-m01 development of infant robot with musculoskeletal and skin system. *The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec)*, 2011:_2P2–M01_1–_2P2–M01_4. Cited on page 98.

Nemec, B., L. Žlajpah, and A. Ude
2017. Door opening by joining reinforcement learning and intelligent control. In *18th International Conference on Advanced Robotics (ICAR)*, Pp. 222–228. Cited on page 1.

Neumann, K.
2014. *Reliability of Extreme Learning Machines*. PhD thesis. Cited on page 98.

Neumann, K., M. Rolf, and J. J. Steil
2013. Reliable integration of continuous constraints into extreme learning machines. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 21(Suppl 2):35–50. Cited on pages 98, 100, and 101.

Neumann, K. and J. J. Steil
2013. Optimizing Extreme Learning Machines via Ridge Regression and Batch Intrinsic Plasticity. *Neurocomputing*, 102(Special Issue: Advances in Extreme Learning Machines (ELM 2011)):23–30. Cited on page 27.

Ng, A. Y. and S. J. Russell
2000. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, Pp. 663–670, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Cited on page 45.

Nguyen-Tuong, D. and J. Peters
2010. Using model knowledge for learning inverse dynamics. In *International Conference on Robotics and Automation (ICRA)*, Pp. 2677–2682. IEEE. Cited on page 126.

Nguyen-Tuong, D. and J. Peters
2011. Model learning for robot control: a survey. *Cognitive Processing*, 12(4):319–340. Cited on page 126.

Norrloff, M. and S. Gunnarsson
2002. Experimental comparison of some classical iterative learning control algorithms. *IEEE transactions on robotics and automation*, 18(4):636–641. Cited on pages 35 and 131.

Ogawa, K., K. Narioka, and K. Hosoda
2011. Development of whole-body humanoid pneumat-bs with pneumatic musculoskeletal system. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pp. 4838–4843. Cited on page 98.

Ott, C., B. Henze, and D. Lee
2013. Kinesthetic teaching of humanoid motion based on whole-body compliance control with interaction-aware balancing. In *International Conference on Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ*, Pp. 4615–4621. Cited on page 17.

Paaßen, B., A. Schulz, J. Hahne, and B. Hammer
2018. Expectation maximization transfer learning and its application for bionic hand prostheses. *Neurocomputing*, (298):122–133. Cited on pages 6, 86, and 87.

Paaßen, B., A. Schulz, and B. Hammer
2016. Linear supervised transfer learning for generalized matrix lvq. In *Proceedings of the Workshop New Challenges in Neural Computation (NC^2) 2016*, B. Hammer, T. Martinetz, and T. Villmann, eds., number 4, Pp. 11–18, Germany. Cited on page 87.

Pan, S. J. and Q. Yang
2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359. Cited on pages 86 and 87.

Pao, Y.-H., G.-H. Park, and D. J. Sobajic
1994. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing*, 6(2):163–180. Cited on page 25.

Parisi, S., M. Pirotta, and J. Peters
2017. Manifold-based multi-objective policy search with sample reuse. *Neurocomputing*, 263:3–14. Cited on page 84.

Park, F. C. and K. Jo
2004. *Movement Primitives and Principal Component Analysis*, Pp. 421–430. Dordrecht: Springer Netherlands. Cited on page 66.

Paskarbeit, J., S. Annunziata, and A. Schneider
2013. A resilient robotic actuator based on an integrated sensorized elastomer coupling. In *Proceedings of the 16th International Conference on Climbing and Walking Robots*, Pp. 257–264. Cited on page 16.

Pastor, P., M. Kalakrishnan, F. Meier, F. Stulp, J. Buchli, E. Theodorou, and S. Schaal
2013. From dynamic movement primitives to associative skill memories. *Robotics and Autonomous Systems*, 61(4):351–361. Models and Technologies for Multi-modal Skill Training. Cited on page 19.

Petrič, T., L. Colasanto, A. Gams, A. Ude, and A. J. Ijspeert
2015. Bio-inspired learning and database expansion of compliant movement primitives. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots*, Pp. 346–351. Cited on page 127.

Pfeifer, R. and J. C. Bongard
2006. *How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books)*. The MIT Press. Cited on page 17.

Pfeifer, R. and G. Gómez
2009. Creating brain-like intelligence. chapter Morphological Computation — Connecting Brain, Body, and Environment, Pp. 66–83. Berlin, Heidelberg: Springer-Verlag. Cited on page 18.

Pirotta, M., S. Parisi, and M. Restelli
2015. Multi-objective reinforcement learning with continuous pareto frontier approximation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, Pp. 2928–2934. AAAI Press. Cited on page 84.

Popić, S. and B. Miloradović
2015. Light weight robot arms - an overview. *17th International Symposium INFOTEH-JAHORINA*, 14:818–822. Cited on page 119.

Prahm, C., B. Paaßen, A. Schulz, B. Hammer, and O. Aszmann
2016. Transfer learning for rapid re-calibration of a myoelectric prosthesis after electrode shift. In *Converging Clinical and Engineering Research on Neurorehabilitation II: Proceedings of the 3rd International Conference on NeuroRehabilitation (ICNR2016), October 18-21, 2016, Segovia, Spain*, J. Ibáñez, J. González-Vargas, J. M. Azorín, M. Akay, and J. L. Pons, eds., Pp. 153–157. Springer International Publishing. Cited on page 6.

Pratt, G. A. and M. M. Williamson
1995. Series elastic actuators. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 1, Pp. 399–406 vol.1. Cited on page 16.

Queißer, J. F., H. Ishihara, B. Hammer, J. J. Steil, and M. Asada
2018. Skill memories for parameterized dynamic action primitives on the pneumatically driven humanoid robot child affetto. In *Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Tokyo, Japan. IEEE.

Queißer, J. F., K. Neumann, M. Rolf, R. F. Reinhart, and J. J. Steil
2014. An active compliant control mode for interaction with a pneumatic soft robot. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pp. 573–579.

Queißer, J. F., R. F. Reinhart, and J. J. Steil
2016. Incremental bootstrapping of parameterized motor skills. In *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Pp. 223–229. Cited on pages 23, 40, 64, 126, 127, and 128.

Queißer, J. F. and J. J. Steil
2016. Incremental bootstrapping of parametrized skill memories. In *DGR Days 2016*, D. D. 2016, ed., P. 14.

Queißer, J. F. and J. J. Steil
2018. Bootstrapping of parameterized skills through hybrid optimization in task and policy spaces. *Frontiers in Robotics and AI*, 5(49).

Reinhart, R. F.
2011. *Reservoir Computing With Output Feedback*. PhD thesis. Cited on pages 28 and 30.

Reinhart, R. F.
2017. Autonomous exploration of motor skills by skill babbling. *Autonomous Robots*, 41(7):1521–1537. Cited on pages 23 and 151.

Reinhart, R. F., Z. Shareef, and J. J. Steil
2017a. Hybrid analytical and data-driven modeling for feed-forward robot control. *Sensors*, 17(2). Cited on page 5.

Reinhart, R. F., Z. Shareef, and J. J. Steil
2017b. Hybrid analytical and data-driven modeling for feed-forward robot control. *Sensors*, 17(2):311. Cited on page 126.

Reinhart, R. F. and J. J. Steil
2011. Neural learning and dynamical selection of redundant solutions for inverse kinematic control. In *11th IEEE-RAS International Conference on Humanoid Robots*, Pp. 564–569. Cited on page 30.

Reinhart, R. F. and J. J. Steil
2012. Learning whole upper body control with dynamic redundancy resolution in coupled associative radial basis function networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Pp. 1487–1492. IEEE. Cited on page 30.

Reinhart, R. F. and J. J. Steil
2014. Efficient policy search with a parameterized skill memory. In *IEEE/RSJ*

International Conference on Intelligent Robots and Systems, Pp. 1400–1407. IEEE. Cited on pages 4, 30, 126, and 127.

Reinhart, R. F. and J. J. Steil
2015. Efficient policy search in low-dimensional embedding spaces by generalizing motion primitives with a parameterized skill memory. *Autonomous Robots*, 38(4):331–348. Cited on pages 4, 19, 32, and 130.

Rohmer, E., S. P. N. Singh, and M. Freese
2013. V-rep: A versatile and scalable robot simulation framework. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Pp. 1321–1326. Cited on page 132.

Rolf, M., K. Neumann, J. F. Queißer, F. Reinhart, A. Nordmann, and J. J. Steil
2015. A multi-level control architecture for the bionic handling assistant. *Advanced Robotics*, 29(13: SI):847–859.

Rolf, M. and J. J. Steil
2012. Constant curvature continuum kinematics as fast approximate model for the bionic handling assistant. In *IEEE/RSJ International Conference Intelligent Robots and Systems*, Pp. 3440–3446. Cited on page 17.

Rolf, M. and J. J. Steil
2014. Efficient exploratory learning of inverse kinematics on a bionic elephant trunk. *IEEE Transactions on Neural Networks and Learning Systems*, 25(6):1147–1160. Cited on page 17.

Rolf, M., J. J. Steil, and M. Gienger
2010. Goal babbling permits direct learning of inverse kinematics. *IEEE Transactions Autonomous Mental Development*, 2(3):216–229. Cited on pages 23 and 151.

Romeres, D., M. Zorzi, R. Camoriano, and A. Chiuso
2016. Online semi-parametric learning for inverse dynamics modeling. In *IEEE 55th Conference on Decision and Control*, Pp. 2945–2950, Las Vegas, US. Cited on page 126.

Roover, D. D. and O. H. Bosgra
2000. Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system. *International Journal of Control*, 73(10):968–979. Cited on pages 36 and 131.

Roozing, W., Z. Li, D. G. Caldwell, and N. G. Tsagarakis
2016. Design optimisation and control of compliant actuation arrangements in articulated robots for improved energy efficiency. *IEEE Robotics and Automation Letters*, 1(2):1110–1117. Cited on page 3.

Rutishauser, S., A. Sproewitz, L. Righetti, and A. J. Ijspeert
2008. Passive compliant quadruped robot using central pattern generators for locomotion control. *2008 IEEE International Conference on Biomedical Robotics and Biomechatronics*, Pp. 710–715. Cited on page 16.

Salaken, S. M., A. Khosravi, T. Nguyen, and S. Nahavandi
2017. Extreme learning machine based transfer learning algorithms. *NeuroComputation*, 267(C):516–524. Cited on page 86.

Salakhutdinov, R. and G. Hinton
2009. Deep boltzmann machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 5, Pp. 448–455. PMLR. Cited on page 24.

Schaal, S.
2006. *Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics*, Pp. 261–280. Tokyo: Springer Tokyo. Cited on pages 31 and 32.

Schieber, M. H.
2000. New views of the primary motor cortex. *The Neuroscientist*, 6(5):380–389. Cited on page 18.

Schilling, M., J. Paskarbeit, T. Hoinville, A. Hüffmeier, A. Schneider, J. Schmitz, and H. Cruse
2013. A hexapod walker using a heterarchical architecture for action selection. *Front. in Computational Neuroscience*, 7:126. Cited on page 17.

Schmidt, R. A.
1975. A schema theory of discrete motor skill learning. 82(4):225–260. Cited on pages 4 and 15.

Schmidt, W. F., M. A. Kraaijveld, and R. P. W. Duin
1992. Feedforward neural networks with random weights. In *Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems*, Pp. 1–4. Cited on page 25.

Schneider, A., J. Paskarbeit, M. Schäffersmann, and J. Schmitz
2011. HECTOR, a new hexapod robot platform with increased mobility - control approach, design and communication. In *Advances in Autonomous Mini Robots*, J. Sitte and U. Rückert, eds., Pp. 249–264. Cited on page 17.

Schneider, A., J. Paskarbeit, M. Schilling, and J. Schmitz
2014. HECTOR, a bio-inspired and compliant hexapod robot. In *Proceedings of the 3rd Conference on Biomimetics and Biohybrid Systems*, Pp. 427–430. Cited on page 16.

Schulz, A., J. F. Queißer, H. Ishihara, and M. Asada
2018. Transfer learning of complex motor skills on the humanoid robot affetto. In *Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE.

Schweighofer, N., M. A. Arbib, and M. Kawato
1998. Role of the cerebellum in reaching movements in humans. i. distributed inverse dynamics control. *The European Journal of Neuroscience*, 10(1):86–94. Cited on pages 18 and 127.

Scott, S. H.
2004. Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience*, 5(7):532–546. Cited on page 19.

Scott, S. H.
2008. Inconvenient truths about neural processing in primary motor cortex. *The Journal of Physiology*, 586(5):1217–1224. Cited on page 19.

Seok, S., C. D. Onal, K.-J. Cho, R. J. Wood, D. Rus, and S. Kim
2013. Meshworm: A peristaltic soft robot with antagonistic nickel titanium coil actuators. *IEEE Transactions Mechatronics*, 18(5):1485–1497. Cited on page 17.

Shareef, Z., P. Mohammadi, and J. J. Steil
2016. Improving the inverse dynamics model of the kuka lwr iv+ using independent joint learning. In *Proceedings 7th IFAC Symposium on Mechatronic Systems*, volume 49/21, Pp. 507–512. Elsevier. Cited on page 5.

Sherrington, C. S.
1906. *The Integrative Action of the Nervous System*. New Haven, CT, US: Yale University Press. Cited on page 14.

Shirai, T., J. Urata, Y. Nakanishi, K. Okada, and M. Inaba
2011. Whole body adapting behavior with muscle level stiffness control of tendon-driven multijoint robot. In *IEEE International Conference Robotics and Biomimetics*, Pp. 2229–2234. Cited on page 17.

Sigaud, O. and F. Stulp
2018. Policy search in continuous action domains: an overview. *ArXiv e-prints*. Cited on pages 33 and 40.

Silva, B. D., G. Baldassarre, G. Konidaris, and A. Barto
2014. Learning parameterized motor skills on a humanoid robot. In *IEEE International Conference Robotics and Automation (ICRA)*, Pp. 5239–5244. Cited on pages 4, 19, 22, 23, 40, 126, and 127.

Silva, B. D., GeorgeKonidaris, A. G. Barto, and B. Castro
2012. Learning parameterized skills. In *International Conference on Machine*

Learning (ICML 2012), Pp. 1679–1686. Cited on pages 4, 19, 22, 23, 40, 125, and 127.

Smolensky, P.
1986. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Information Processing in Dynamical Systems: Foundations of Harmony Theory, Pp. 194–281. Cambridge, MA, USA: MIT Press. Cited on page 29.

Soria-Olivas, E., J. Gomez-Sanchis, J. D. Martin, J. Vila-Frances, M. Martinez, J. R. Magdalena, and A. J. Serrano
2011. Belm: Bayesian extreme learning machine. *IEEE Transactions on Neural Networks*, 22(3):505–509. Cited on page 27.

Spivak, M.
1971. *Calculus on Manifolds: A Modern Approach to Classical Theorems of Advanced Calculus*, chapter 2.5, Pp. 34–40. Westview Press. Cited on page 74.

Spröwitz, A., A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert
2013. Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot. *International Journal of Robotics Research*, 32(8):932–950. Cited on page 17.

Stulp, F., E. Oztop, P. Pastor, M. Beetz, and S. Schaal
2009. Compact models of motor primitive variations for predictable reaching and obstacle avoidance. In *9th IEEE-RAS International Conference on Humanoid Robots*, Pp. 589–595. Cited on page 66.

Stulp, F., G. Raiola, A. Hoarau, S. Ivaldi, and O. Sigaud
2013. Learning compact parameterized skills with a single regression. In *IEEE-RAS International Conference on Humanoid Robots*, Pp. 417–422. Cited on pages 19 and 22.

Stulp, F. and O. Sigaud
2013. Policy improvement: Between black-box optimization and episodic reinforcement learning. In *Journées Francophones Planification, Décision, et Apprentissage pour la conduite de systèmes*. Cited on pages 4, 33, 44, and 68.

Stulp, F. and O. Sigaud
2015. Many regression algorithms, one unified model: A review. *Neural Networks*, 69:60 – 79. Cited on page 24.

Taylor, M. E. and P. Stone
2009. Transfer learning for reinforcement learning domains: A survey. *J. Mach. Learn. Res.*, 10:1633–1685. Cited on page 87.

Theodorou, E. A., J. Buchli, and S. Schaal
2010. Learning policy improvements with path integrals. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTAT)*, Pp. 828–835. Cited on page 44.

Thorndike, E. L.
1898. Animal intelligence: An experimental study of the associative processes in animals. *The Psychological Review: Monograph Supplements*, 2(4):i–109. Cited on page 14.

Tichonov, A. N. and V. J. Arsenin
1977. *Solutions of Ill-Posed Problems*, Scripta series in mathematics. Washington, DC: Winston. Cited on page 27.

Todorov, E., C. Hu, A. Simpkins, and J. Movellan
2010. Identification and control of a pneumatic robot. In *3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Pp. 373–380. Cited on pages xiii, 59, 110, 114, 117, 129, and 135.

Tondu, B., S. Ippolito, J. Guiochet, and A. Daidie
2005. A seven-segrees-of-freedom robot-arm driven by pneumatic artificial muscles for humanoid robots. *The International Journal of Robotics Research*, 24(4):257–274. Cited on page 17.

Toussaint, M. and M. G. C. Goerick
2007. Optimization of sequential attractor-based movement for compact behaviour generation. In *IEEE-RAS International Conference on Humanoid Robots*, Pp. 122–129. Cited on page 55.

Transeth, A. A., R. I. Leine, C. Glocker, K. Y. Pettersen, and P. Liljebäck
2008. Snake robot obstacle-aided locomotion: Modeling, simulations, and experiments. *IEEE Transactions Robotics*, 24(1):88–104. Cited on page 17.

Tsagarakis, N., Z. Li, J. A. Saglia, and D. G. Caldwell
2011. *The Design of the Lower Body of the Compliant Humanoid Robot 'cCub'*, Pp. 2035–2040. IEEE. Cited on page 17.

Tsagarakis, N. G., M. Laffranchi, B. Vanderborght, and D. G. Caldwell
2009. A compact soft actuator unit for small scale human friendly robots. In *2009 IEEE International Conference on Robotics and Automation*, Pp. 4356–4362. Cited on page 17.

Ude, A., A. Gams, T. Asfour, and J. Morimoto
2010. Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Transactions on Robotics*, 26(5):800–815. Cited on page 4.

Ude, A., M. Riley, B. Nemec, A. Kos, T. Asfour, and G. Cheng
2007. Synthesizing goal-directed actions from a library of example movements. In *IEEE-RAS International Conference on Humanoid Robots*, Pp. 115–121. Cited on page 19.

Ullah, A. S. S. M. B., R. Sarker, and D. Cornforth
2008. Search space reduction technique for constrained optimization with tiny feasible space. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, GECCO '08, Pp. 881–888, New York, NY, USA. ACM. Cited on page 66.

Verrelst, B., R. V. Ham, B. Vanderborght, D. Lefeber, F. Daerden, and M. V. Damme
2006. Second generation pleated pneumatic artificial muscle and its robotic applications. *Advanced Robotics*, 20(7):783–805. Cited on page 17.

Vuong, N. D. and M. H. A. Jr.
2009. Dynamic model identification for industrial robots. *Acta Polytechnica Hungarica*, 6(5):51–68. Cited on page 119.

Wang, C. and S. Mahadevan
2008. Manifold alignment using procrustes analysis. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, Pp. 1120–1127, New York, NY, USA. ACM. Cited on page 87.

Wang, W., R. N. Loh, and E. Y. Gu
1998. Passive compliance versus active compliance in robot-based automated assembly systems. *Industrial Robot: An International Journal*, 25(1):48–57. Cited on page 104.

Wang, Y., F. Gao, and F. J. Doyle
2009. Survey on iterative learning control, repetitive control, and run-to-run control. *Journal of Process Control*, 19(10):1589 – 1600. Cited on pages 35 and 36.

Wersing, H. and E. Körner
2003. Learning optimized features for hierarchical models of invariant object recognition. *Neural Computation*, 15(7):1559–1588. Cited on page 24.

Whelan, P. J.
1996. Control of locomotion in the decerebrate cat. *Progress in Neurobiology*, 49(5):481 – 515. Cited on page 18.

Whitney, J. P., M. F. Glisson, E. L. Brockmeyer, and J. K. Hodgins
2014. A low-friction passive fluid transmission and fluid-tendon soft actuator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pp. 2801–2808. Cited on pages 95 and 126.

Williams, R. J.
1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256. Cited on pages 41 and 44.

Wolf, S., G. Grioli, O. Eiberger, W. Friedl, M. Grebenstein, H. Höppner, E. Burdet, D. G. Caldwell, R. Carloni, M. G. Catalano, et al.
2016. Variable stiffness actuators: Review on design and components. *IEEE/ASME transactions on mechatronics*, 21(5):2418–2430. Cited on page 17.

Wolpert, D. M. and Z. Ghahramani
2000. Computational principles of movement neuroscience. *Nature Neuroscience*, 3:1212–1217. Cited on page 18.

Woodworth, R. S.
1899. Accuracy of voluntary movement. *The Psychological Review: Monograph Supplements*, 3(3):i–114. Cited on page 14.

Woodworth, R. S. and E. L. Thorndike
1901. The influence of improvement in one mental function upon the efficiency of other functions.(i). *Psychological review*, 8(3):247–261. Cited on page 14.

Wrede, S., C. Emmerich, R. Grünberg, A. Nordmann, A. Swadzba, and J. J. Steil
2013. A user study on kinesthetic teaching of redundant robots in task and configuration space. *Journal of Human-Robot Interaction*, 2(1):56–81. Cited on page 18.

Wundt, W. M., J. E. Creighton, and E. B. Titchener
1907. *Lectures on Human and Animal Psychology*. William Swan Sonnenschein and Macmillan & Company. Translated from the Second German Edition (1841). Cited on page 14.

Yin, Y., Z. Guo, X. Chen, and Y. Fan
2012. Studies on biomechanics of skeletal muscle based on the working mechanism of myosin motors: An overview. *Chinese Science Bulletin*, 57(35):4533–4544. Cited on page 3.

Zhao, J., Z. Wang, and D. S. Park
2012. Online sequential extreme learning machine with forgetting mechanism. *Neurocomputing*, 87:79–89. Cited on page 44.

Zinn, M., O. Khatib, B. Roth, and J. K. Salisbury
2004. A new actuation approach for human friendly robot design. *International Journal of Robotics Research*, 23(4–5):379–398. Cited on pages xii, 16, 95, and 126.