

# **A Continuous Grasp Representation for the Imitation Learning of Grasps on Humanoid Robots**

zur Erlangung des akademischen Grades eines  
Doktors der Ingenieurwissenschaften

von der Fakultät für Informatik  
des Karlsruher Instituts für Technologie (KIT)

**genehmigte**

**Dissertation**

von

**Martin Minh Thong Do**

aus Aalen

Tag der mündlichen Prüfung: 12.06.2014

Erster Gutachter: Herr Prof. Dr.-Ing.Rüdiger Dillmann

Zweiter Gutachter: Herr Prof. Dr. Erhan Oztop



## Acknowledgment

This thesis was carried out in the course of my employment as research assistant at the High Performance Humanoid Technologies Lab (H<sup>2</sup>T) of the Institute for Anthropomatics and Robotics (IAR), Karlsruhe Institute of Technology (KIT). First of all I want to thank my doctoral supervisor Prof. Dr.-Ing. Rüdiger Dillmann for giving me the opportunity to work on this fascinating topic. I want to thank Prof. Dillmann for his support during the last years. I am very grateful to Prof. Dr. Erhan Oztop for his interest in my work, for his valuable advice, and for joining the committee as co-supervisor.

My deepest gratitude is extended to Prof. Dr.-Ing. Tamim Asfour, head of the H<sup>2</sup>T lab, for providing the environment and the humanoid platforms which have been crucial for my work. To this, day, his commitment, his guidance, and his faith in my person light my way. Further, I want to express my gratitude to Dr. Mitsuo Kawato, head of the Computational Neuroscience Laboratories (CNS) of the Advanced Telecommunications Research Institute International (ATR), for the opportunity to work in his lab in 2006 / 2007. Especially, I want to thank Prof. Dr. Gordon Cheng for the supervision during my stay.

I would like to express my gratitude to my colleagues at the H<sup>2</sup>T lab and thank them for the excellent teamwork and the great support. Especially, I want to thank my closest friends Dr. Kai Welke and Dr. Nikolaus Vahrenkamp for being there for me in good and in bad times. I owe my special thanks to Dr. Pedram Azad for his support and his patience in the supervision of my diploma thesis and for the inspiring discussions which kept me on track. My thank also goes to all the other colleagues of the humanoids group: Mirko Wächter, Dr. Julia Borrás Sol, Ömer Terlemez, David Schiebener, Christian Mandery, Fabian Schültje, Manfred Kröhnert, Jonas Beil, Lukas Kaul, Peter Kaiser, Michael Neaga, and Simon Ottenhaus. I also want to thank my former colleagues Dr. David Gonzalez, Dr. Stefan Ulbrich, Dr. Markus Przybylski, Julian Schill, Paul Holz, Stefan Gärtner, Sebastian Schulz, my colleagues of the medicine group, the programming by demonstration group, and the cognitive cars group. Further, I want to thank our secretaries Christine Brand, Diana Becker, and Isabelle Wappler for their help in all situations. Many thanks to my students for their interest in my work, their efforts, and their contributions. In particular, I want to thank Johannes Ernesti, a bright student, an excellent researcher, and a good friend, who enriched my work with his ideas and views.

Finally, I want to thank my parents for their endless support and most importantly my wife Lam Huong and my daughter Mia for their patience and their love.



# Contents

<b>Acknowledgment</b> . . . . .	<b>i</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Motivation and Objective . . . . .	1
1.2 Contributions . . . . .	3
1.3 Outline . . . . .	4
<b>2 Insights in Human Grasping</b> . . . . .	<b>7</b>
2.1 Grasping Hand Shapes . . . . .	7
2.2 Grasping Kinematics . . . . .	8
2.2.1 Postural Hand Synergies . . . . .	12
<b>3 Grasp Representations in Robotics</b> . . . . .	<b>13</b>
3.1 Human-Inspired Discrete Grasp Representations . . . . .	14
3.2 Human-Inspired Continuous Grasp Representations . . . . .	17
3.2.1 Representations in Joint Space . . . . .	18
3.2.2 Representations in Task Space . . . . .	20
3.3 Discussion . . . . .	21
<b>4 Grasp Representation</b> . . . . .	<b>23</b>
4.1 Motivation . . . . .	23
4.1.1 Finger Movement Synergies . . . . .	25
4.2 Basic Principles . . . . .	27
4.2.1 Multi-Body Systems . . . . .	28
4.3 Virtual Spring Grasp Representation . . . . .	29
4.3.1 Contact Springs . . . . .	29
4.3.2 Finger Springs . . . . .	31
4.3.3 Stabilization Springs . . . . .	32
4.3.4 Virtual Contact Strip . . . . .	33
4.3.5 Enclose and Preshape . . . . .	34
4.4 Representation of Transport . . . . .	35
4.4.1 Dynamic Movement Primitives . . . . .	36
4.5 Coupling between Transport and Grip . . . . .	37
4.6 Summary . . . . .	37
<b>5 Grasp Data Acquisition</b> . . . . .	<b>39</b>
5.1 Human Motion Capture with Optical Systems . . . . .	39
5.2 Marker-based Motion Capturing . . . . .	40
5.3 Markerless Motion Capturing . . . . .	41
5.3.1 Advances in Markerless Human Motion Capture . . . . .	41
5.3.2 Particle Filter Framework . . . . .	43
5.3.3 Upper Body Tracking . . . . .	43
5.3.4 Hand Tracking . . . . .	44

5.3.5	Fingertip Tracking . . . . .	45
5.4	Grasp Motion Data . . . . .	51
5.4.1	Segmentation of Grasp Examples . . . . .	52
5.5	Summary . . . . .	53
<b>6</b>	<b>Parameter Estimation for a Grasp Representation . . . . .</b>	<b>55</b>
6.1	Parameter Estimation . . . . .	55
6.2	Problem Statement . . . . .	56
6.2.1	Observational Data . . . . .	56
6.3	Parameter Estimation Scheme . . . . .	57
6.3.1	Solving the TLS Problem . . . . .	60
6.3.2	Initial Solution . . . . .	62
6.3.3	Local Estimation of Spring Constant Parameters . . . . .	62
6.3.4	Nonnegativity Constraints . . . . .	63
6.3.5	Estimation of Data Correction . . . . .	64
6.4	Weight Formation . . . . .	65
6.4.1	Weighted Estimation of Spring Constant Parameters . . . . .	66
6.5	Instantiation of the Dynamic Movement Primitive . . . . .	66
6.6	Summary . . . . .	67
<b>7</b>	<b>Grasp Learning and Execution . . . . .</b>	<b>69</b>
7.1	Related works . . . . .	69
7.2	System Overview . . . . .	70
7.3	Mapping . . . . .	70
7.3.1	Master Motor Map . . . . .	70
7.3.2	Conversion of Fingertip Movements . . . . .	71
7.3.3	Conversion of Arm Movements . . . . .	72
7.4	Structuring of Grasp-Related Motor Knowledge . . . . .	74
7.4.1	Grasp Type Classification . . . . .	74
7.5	Reproduction on the Robot . . . . .	74
7.5.1	Distance-Based Coupling between Transport and Grip . . . . .	75
7.6	Summary . . . . .	76
<b>8</b>	<b>Evaluation . . . . .</b>	<b>79</b>
8.1	Experimental Setup . . . . .	79
8.1.1	Experimental Platforms . . . . .	79
8.1.2	Object Set . . . . .	80
8.2	Grasp Data Acquisition . . . . .	81
8.3	Grasp Modeling . . . . .	83
8.3.1	Grasp Parameter Estimation . . . . .	83
8.4	Grasp Adaptation . . . . .	90
8.4.1	Adaptation of Arm Movements . . . . .	91
8.4.2	Adaptation of Finger Movements . . . . .	92
8.4.3	Behavior under Perturbation . . . . .	95
8.5	Grasp Reproduction . . . . .	96
8.5.1	Reproduction of Arm Movements . . . . .	96
8.5.2	Reproduction of Finger Movements . . . . .	97
8.6	Summary . . . . .	99
<b>9</b>	<b>Conclusion . . . . .</b>	<b>105</b>

---

9.1	Contribution . . . . .	105
9.2	Discussion and Outlook . . . . .	106
<b>A</b>	<b>Libraries and Tools . . . . .</b>	<b>109</b>
A.1	HMC . . . . .	110
A.2	MotionLearning . . . . .	110
A.3	ClaRe . . . . .	111
A.4	Grasp Learning Framework . . . . .	111
<b>B</b>	<b>Learning of Rhythmic Manipulation Actions . . . . .</b>	<b>115</b>
B.1	Representation of Rhythmic Actions . . . . .	115
B.1.1	Encoding of Transient Periodic Movements . . . . .	116
B.1.2	Learning of Rhythmic Actions . . . . .	117
B.1.3	Adaptation to Environment . . . . .	118
B.2	Conclusion . . . . .	120
<b>C</b>	<b>Reasoning of an Action based on Object-Action Affordances . . . . .</b>	<b>123</b>
C.1	Learning of Affordances from Exploration and Physical Interaction . . . . .	123
C.2	The Learning Cycle . . . . .	124
C.2.1	Instantiation of the Learning Cycle for Wiping . . . . .	124
C.2.2	Exploration of Object Features . . . . .	125
C.2.3	Exploration of Action Parameters . . . . .	125
C.2.4	Exploration of Effect . . . . .	126
C.2.5	Learning of Internal Models . . . . .	127
C.3	Experiments . . . . .	128
C.4	Conclusion . . . . .	130





# 1. Introduction

Robots have evolved to indispensable helpers in today's society and have been proven to be one of the most important technological advances in recent human history. Particularly, in the field of manufacturing, one cannot imagine an industrial environment without robots performing tasks such as pick and place operations of heavy objects, high-precision cutting, and welding of materials with such an efficiency which is unattainable for humans. Robots have also spread into our homes, currently, in the form of little helpers taking over simple tasks (such as vacuum cleaning) which are far from being capable of proper interaction with the scene among others due to lack of essential grasping and manipulation abilities.

Towards artificial platforms capable of incorporating these abilities, research efforts in humanoid robotics have been dedicated to the development of sophisticated systems capable of mimicking the functionalities of a human. The progress in this field led to humanoid platforms with huge potential which to some extent has been demonstrated in specific assistance and manipulation scenarios performed in previously known human-centered environments. However, the potential still remains untapped when robots are confined to a specific task and an environment. To clear this hurdle, robotic platforms are to be endowed with cognitive abilities for the acquisition of novel motor knowledge and the adaptation of this knowledge to unseen situations in order to account for dynamic changes. An intuitive way to approach this challenge is to study humans and to transfer knowledge about learning mechanisms and strategies to robots. In the context of action knowledge acquisition, an emerging paradigm is programming by demonstration, which in recent years progressed to the more biological-oriented term of imitation learning. Following this paradigm, various approaches have been proposed combining statistical learning methods and human observation for the extraction and generalization of knowledge about motor skills. However, for an appropriate interaction with objects and scenes, many challenges have to be addressed, specifically regarding the observation of fine motor movements in humans as well as the representation and implementation of corresponding skills which are necessary to induce object- and task-specificity in robotic grasping and manipulation.

## 1.1. Motivation and Objective

Grasping is one of the most essential capabilities that humans need to interact with the environment. The way how the arms and the fingers are moved in order to grasp a specific object depends on a variety of different object-specific properties and the intention that is pursued. For example, lifting heavy objects requires humans to perform grasps featuring firm grips which maximize the forces exerted on the object while light and fragile objects are to be grasped with the minimal amount of force but higher precision. Regarding one's intention, a grasping movement has to be executed according to the constraints imposed by the task. The multitude of purposes associated with a single object yields a variety of different grasp possibilities that requires a high level of dexterity and flexibility. For humans, this level is easily attained thanks to their hands which are complex organs comprising a large number of joints and muscles as well as sophisticated sensing capabilities. To mimic the functionality of the human hand, research conducted in the field of robotic hand design led to elaborate systems theoretically capable of performing dexterous hand movements. However, in practice,

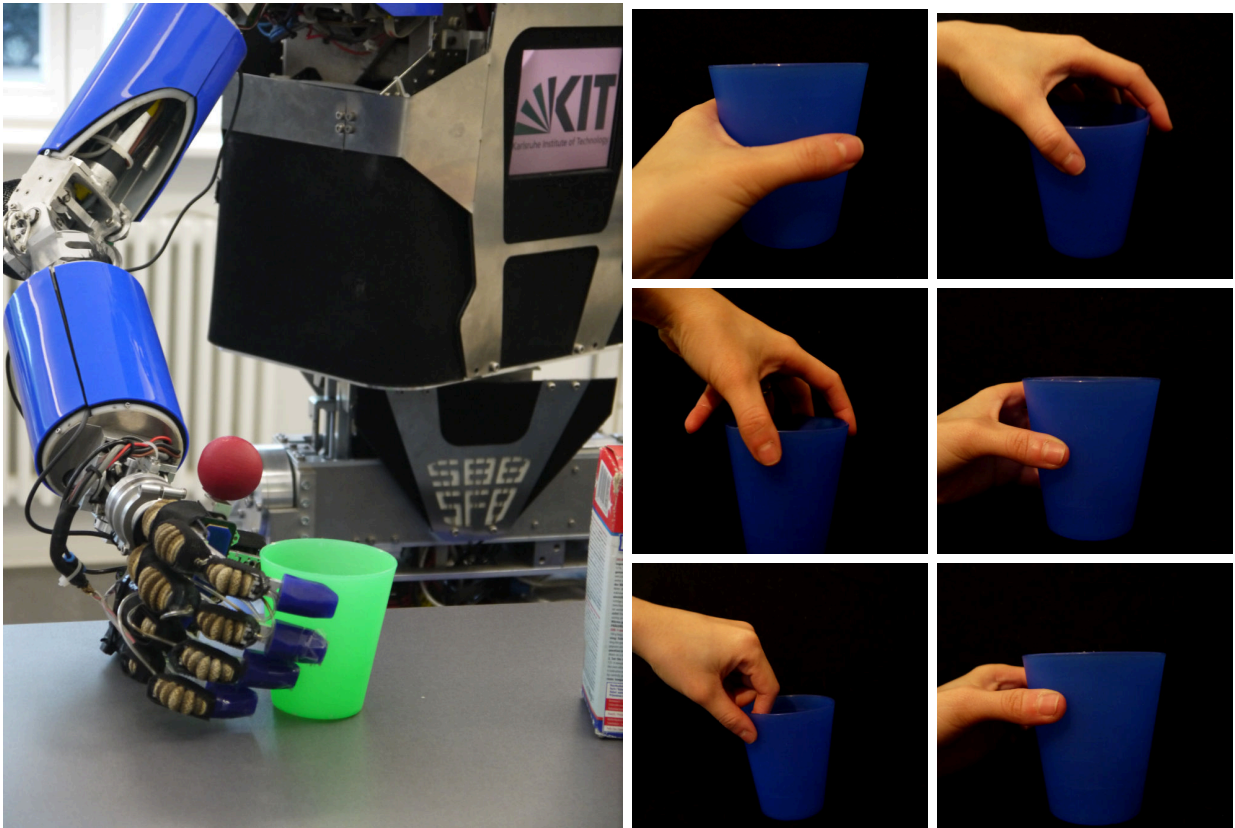


Figure 1.1.: Left: The humanoid robot ARMAR-IIIa (see (Asfour et al., 2006b)) executing a grasping action adjusted to the cup. Right: Human subject performing grasping actions for different tasks such as drinking, placing, passing-over, or carrying.

due to the grasp variety and the complexity of coordinated finger movements the generation and execution of goal-directed grasping and manipulation actions involves the utmost effort. Regarding the human example, to facilitate the synthesis of prehensile movements, the human brain possesses a large repertoire of control mechanisms and motion patterns, and a comprehensive knowledge about the environment which enable the human to efficiently select and perform a wide range of different grasping actions suitable for every situation. This grasp-related motor knowledge is grounded and continuously enriched through extensive training through exploration and observation. The variety of human grasping strategies is exemplified in Figure 1.1 where each grasp is associated with the same object but a different task. So far, the complexity of implementing grasping strategies on robotic platforms requires a strong simplification of the problem by mapping a multitude of different grasping strategies to a single grasping action. Another aspect which increases the complexity of robotic grasping is that with traditional methods implementations grasping strategies have to be confined to specific situations in order to ensure their applicability. Towards autonomous grasping and manipulation in robots, instead of defining and programming grasping actions which can only be executed under certain conditions, the implementation of a cognitive grasping behavior becomes indispensable to make robotic grasping tractable.

The objective of this thesis is to establish a methodology which allows robots to learn grasp primitives from human observation. Therefore, methods and representations have to be developed which enable a robot to obtain grasp-related motor knowledge by extracting and generalizing characteristic features of a grasping movement demonstrated by a human counterpart. Based on these features, movement representations are to be instantiated in

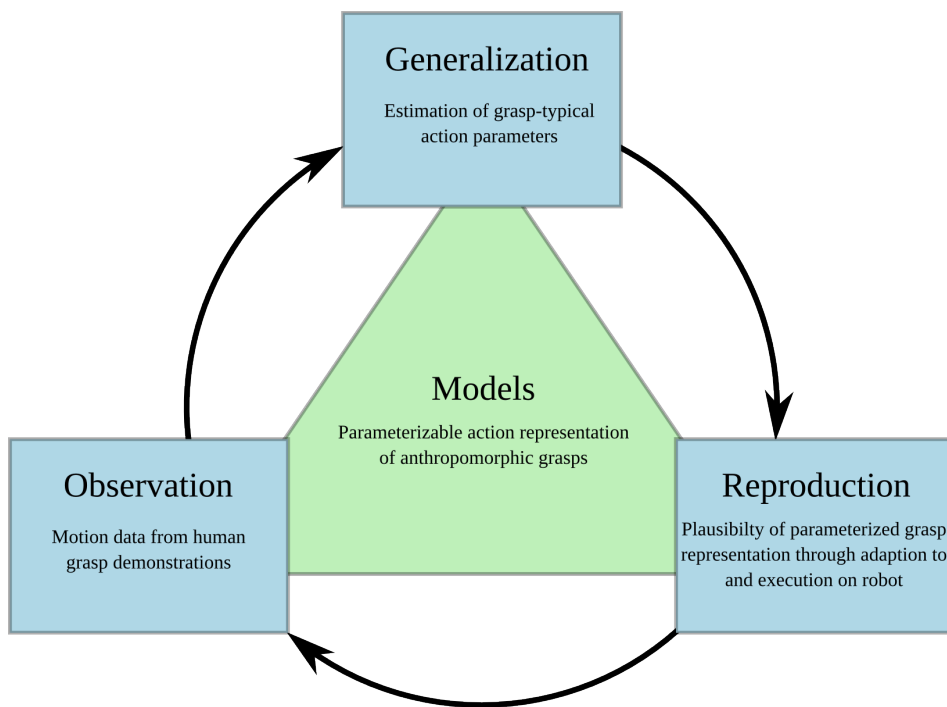


Figure 1.2.: The building blocks of a behavior which allows grasp learning from human demonstration: observation, generalization, and reproduction. The models which specify the grasp representation form the foundation of such behavior. Based on these models the features which have to be observed are determined, the functional relationships between observational features and action parameters are specified, and the structure for the generation of an adapted grasping action is inferred.

order to form grasp primitives which can be adapted and applied to perform grasping in different situations.

## 1.2. Contributions

In this thesis, an approach for the implementation of a learnable and extendable grasping behavior is proposed in order to enhance the grasping capabilities of a humanoid robot. As depicted in Figure 1.2, such a grasp learning behavior can be illustrated as a learning cycle which comprises methods and mechanisms for the observation of human grasps, the generalization of demonstrated grasps, and the reproduction of learned grasp-related motor knowledge. A crucial factor for the success of such a learning cycle lies in the way of how motor knowledge is represented. The models used to describe an action determine the relevant features which need to be observed and extracted from human grasp demonstrations. Based on the features, we want to parameterize these models in a way that a representation is inferred which, on the one hand, generalizes a grasping movement allowing the adaptation to novel situations, and, on the other hand, retains movement characteristics associated with the demonstrated grasp type. To ground an instantiated grasp representation, based on the parameterized models a grasping action has to be generated and executed on the robot. Therefore, the main contribution of this thesis consists of a grasp representation which can serve as a foundation for the described grasp learning behavior. Based on this representation building blocks for the observation, generalization, and reproduction of demonstrated grasps are implemented and combined to a grasp learning framework. Thus, the contributions of this thesis can be summarized as follows:

### • **Continuous Grasp Representation**

A new approach for a continuous grasp representation in the task space is presented. To obtain a complete description of the entire grasping process, beginning with the preshaping of the hand and ending with the enclosing of the object, the representation comprises two motion models of different granularity for the hand and the finger movement. For the representation of the hand movement existing methods capable of encoding reaching movements are studied and evaluated with regard to their applicability in the domain of grasping. Regarding the representation of the finger movements, a novel representation based on virtual springs is proposed. Each fingertip is considered as a single mass point within a dynamical multi-body system where mass spring damper systems between the fingers are used to emulate finger movement synergies. It has been shown that these synergies play an essential role in the reduction of the control complexity for dexterous grasping in humans. In order to ensure that prehensile finger movements are represented in a goal-directed way, designated contact locations are interpreted as attractor points which form an equilibrium state to which the dynamical system converges.

### • **Learning of Grasps from Human Observation**

The observation of human grasp demonstrations is accomplished by various human motion capture methods. For the learning of basic grasp primitives, a marker-based system is employed in order to obtain accurate motion data. In the efforts to increase the autonomy of the robot, markerless methods are investigated and developed for the position-invariant capturing of the human movements using the robot's onboard stereo camera system. In this context, an existing upper body tracking framework is extended by a visual tracking procedure which allows the capturing of prehensile finger movements based on circular image features and by means of stochastic methods for state estimation.

Based on the recorded grasping movements the parameters needed for the instantiation of the grasp are estimated using a nonlinear parameter estimation scheme. The instantiated grasp is labeled according the observed grasp type and stored for later use.

The presented grasp representation and the corresponding grasp learning framework are implemented and evaluated on a humanoid platform. The main goal of the evaluation is the application of the proposed approach for the learning of grasp primitives representing common human grasp types as specified in established grasp taxonomies. In real world scenarios, these primitives are adapted to altering object- and task-specific conditions in order to derive goal-directed grasping actions.

### **1.3. Outline**

A central point in this work is the transfer of human grasp knowledge and mechanisms to robots. For a better understanding, relevant studies conducted on human grasping are reviewed in Chapter 2. A major focus is set on models which explain human grasp kinematics and, thus, can be used to derive a representation for grasping actions. In Chapter 3, grasp representations which have been proposed in previous works are discussed. Of particular interest are approaches which exploit findings in human grasping in order to facilitate grasping on anthropomorphic platforms. Subsequently, in Chapter 4, the proposed grasp representation is presented which allows the encoding of prehensile finger and hand approach movements. First, the principles which form the basis of the representation are formalized in order to

introduce the terminology used in this thesis and to provide a clearer understanding of the approach.

As previously mentioned, to enable a robot to learn grasps from human demonstration, three different subproblems have to be addressed: the observation, generalization, and the reproduction. Methods which allow the generalization of grasp examples to adaptable grasp primitives are subject of Chapter 6. In order to generate grasp examples, in Chapter 5, ways are discussed which enable the acquisition of grasp-relevant motion data from human observation. Based on the developed methods, components are implemented and integrated in a grasp learning framework. An overview of the framework as well as details on the implementation and execution on robots are provided in Chapter 7. Chapter 8 discusses how a grasp can be learned and synthesized using the developed framework. This is determined by evaluating the proposed grasp representation. Finally, the contributions of the thesis are summarized in Chapter 9 and future developments are briefly discussed.



## 2. Insights in Human Grasping

Towards more demanding grasping and manipulation tasks with highly-complex robotic systems such as humanoid robots, the need for grasp representations arises with which the control complexity of these systems can be reduced and which allow the implementation of reusable skills. Hence, in recent years, findings emerging from studies conducted on human grasping take on greater significance in the design process of robotic grasp representations, since these give insights on the underlying functional principles and mechanisms that determine human grasping behavior and provide information on how dexterous grasping behavior can be implemented. In this chapter, studies on human grasping and the modeling of grasping kinematics are reviewed.

### 2.1. Grasping Hand Shapes

The shape of the hand has a tremendous effect on the location and the size of the contact areas as well as the forces applied on the object. Hence, to attain a suitable grasp a hand posture has to be formed satisfying the stability constraints imposed by the object properties and allowing the execution of a subsequent task. As stated in (Castiello, 2005), to facilitate the control of the hand, the human brain is equipped with a repertoire of different grasp postures, each applicable under specific constraints. These grasp postures, respectively hand shapes, are commonly known. Based on anticipated object properties and the task that is pursued, the human brain selects a suitable hand shape to derive the final grasp.

A very early classification of different hand postures in grasping is introduced in (Schlesinger, 1919). Based on the observation of how different objects are grasped and held, six basic grasp types are identified: spherical, palmar, tip, lateral, and hook prehension. The grasp types considered in the classification have been categorized based on different object geometries while any task-specific information has been disregarded. A more task-related categorization is provided by (Napier, 1956). The observed grasp postures were used to denote grasping and manipulation tasks performed in the context of industrial manufacturing. These postures are coarsely divided into two categories, power and precision grasps, where the main difference between power and precision grasps lies in the involvement of the palm. Power grasps feature palm contact, whereas precision grasps merely involve the thumb and the finger areas in order to attain a stable grasp posture. In (Kamakura, 1980), by observing several human subjects grasping and holding daily life objects 14 different grasp types have been identified. As depicted in Figure 2.1, these are subdivided into four different categories: power grip, intermediate grip, precision grip, adduction grip. Power grips are grasp types which are characterized by the involvement of the palm as well. Grasps without palm involvement are either assigned to the intermediate grip or the precision grip category depending on the extent of the finger flexion movement. A special class is described by the adduction grip category which subsumes grasps without thumb involvement. Each grasp type is categorized according to the different contact areas of hand and fingers with the grasped object. Hence, object geometry and the task associated with the object play a very essential role in the definition of a grasp type.

A comprehensive and widely accepted human grasp taxonomy is presented in (Cutkosky, 1989). Based on extensive observations and interviews of humans performing single-handed

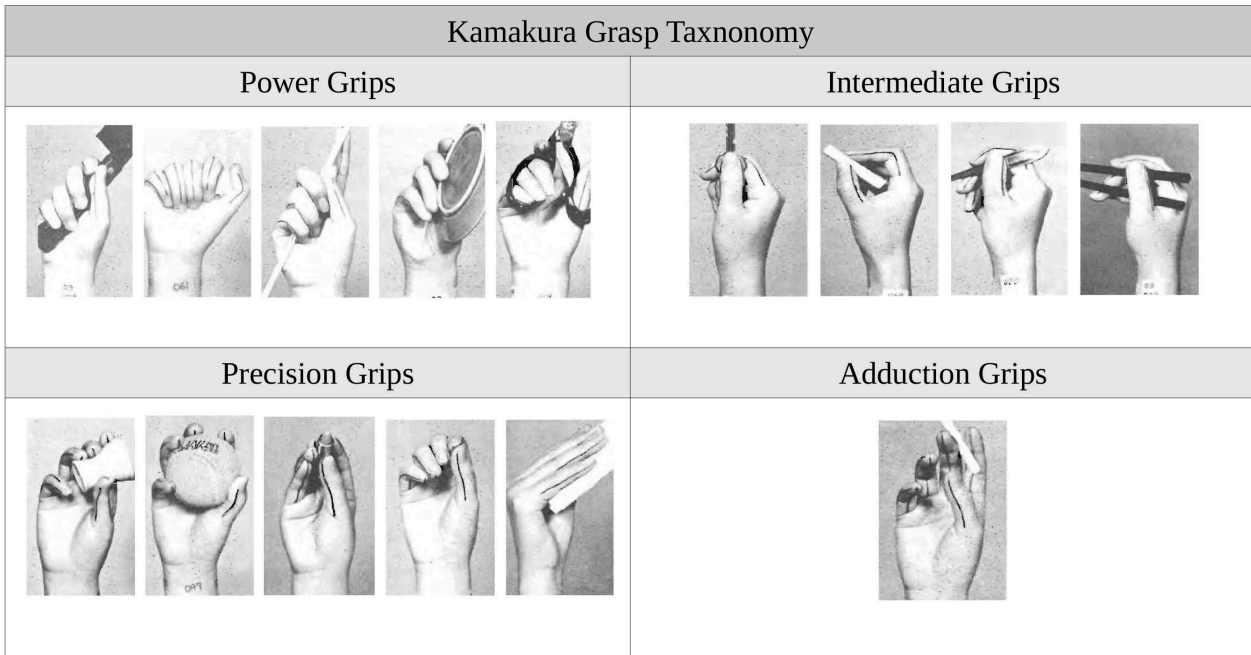


Figure 2.1.: Illustration of the Kamakura taxonomy of prehension as introduced in (Kamakura, 1980). Power grips require palm contact with the object in order to maximize the forces and, thus, the stability of the grasps. In intermediate grips, by maximizing the contact areas between the fingertips and the object, the object is fixated. Grasp types focusing merely on dexterity are subsumed in the precision grip class where minimal forces are exerted by the fingertips. The adduction grip is characterized by fixation of an object using the fingertip sides without involvement of the thumb.

industrial manufacturing tasks 16 different grasp types have been identified. The choice which grasp type to use depends on the precision and the forces which are necessary to perform a certain task with a specific object. Motivated by the categorization originating from (Napier, 1956), as depicted in Figure 2.2, the grasp types are arranged in a tree-like hierarchy where the most-left grasp type, the heavy wrap, represents the most powerful but least dexterous grasp and the most-right grasp type, the thumb-index-finger pinch grasp, features maximum precision.

A recent work on grasp taxonomy is presented in (Feix et al., 2009). By combining various grasp taxonomies proposed in existing literature 45 different grasp types have been identified from which 33 grasp types are considered to satisfy grasp stability constraints. The proposed taxonomy suggests a classification of grasps based on power and precision requirements as well as the oppositional configuration featured by the corresponding hand posture.

## 2.2. Grasping Kinematics

The kinematics of human grasping has been extensively studied leading to a number of models which attempt to explain principles and mechanisms that describe human grasping behavior. A common simplification lies in the assumption that the hand aperture movement of the fingers can be reduced to the actions of two functional units. This view emerges from the virtual finger concept introduced in (Arbib et al., 1985) which postulates that multiple physical fingers can be represented as a single imaginary unit, the virtual finger. It is assumed that in order to grasp objects at least two virtual fingers, mostly thumb and index finger, apply forces on the object surface in opposition to each other. In (Iberall, 1986), this assumption



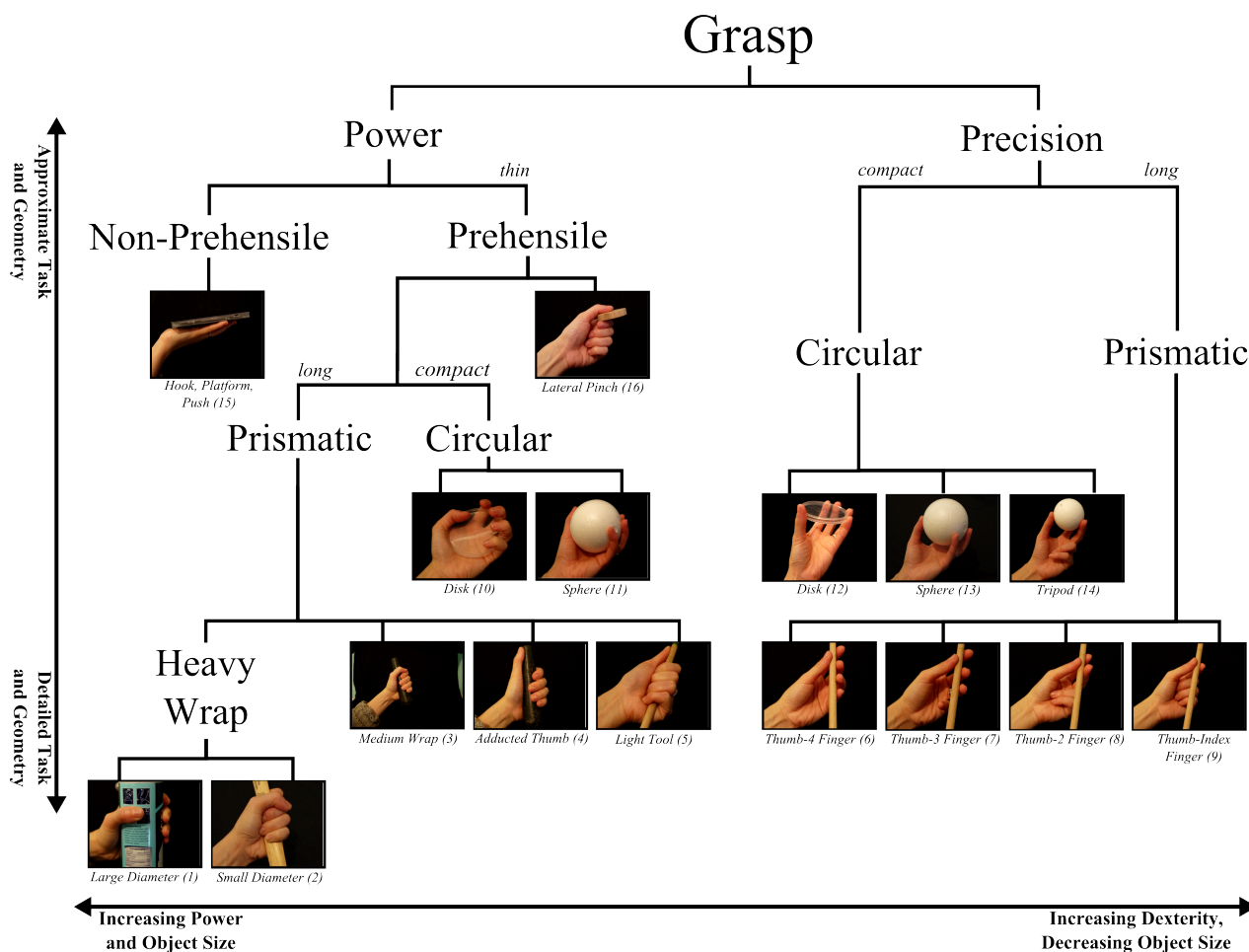


Figure 2.2.: Illustration of the Cutkosky grasp taxonomy (Cutkosky, 1989). The hand postures reflect common grasps applied in manufacturing tasks and are ordered vertically according to the details of the task and object information and horizontally depending on the dexterity needed to perform a grasp. (©1989 IEEE.)

is supported by a comprehensive study on oppositional hand configurations, which comprises additional contact areas of the hand such as the palm. Forces in opposition can be attained by applying three basic grasp types: pad opposition, palm opposition, and side opposition. The difference between these grasp types lies in the direction of the applied forces with regard the hand's palm. Grasps with forces exerted parallel to the palm are categorized as pad opposition. In palm opposition, forces occur between hand surfaces perpendicular to the palm which is the case when fingers are wrapped around an object, for example when grasping a hammer. In side opposition, forces are applied between the thumb and the side of the index finger, transverse to the palm.

A major focus in the literature has been the coordination between the reach and the hand aperture movement. A fundamental view suggested by (Jeannerod, 1981) is based on the existence of two independent components for the control of the limbs during grasping. It is assumed that a transport component is responsible for moving the wrist towards a target object, while a grip component controls the preshaping and enclosing of the fingers. Within the grip component, the fingers are viewed as a single unit, hence, the finger movements are assumed to be determined by the hand aperture which describes the distance between the thumb and the index finger. The independence of both components is based on the hypothesis of two independent visuomotor channels derived from the observation that the hand transport merely depends on extrinsic object properties such as position and orientation within the

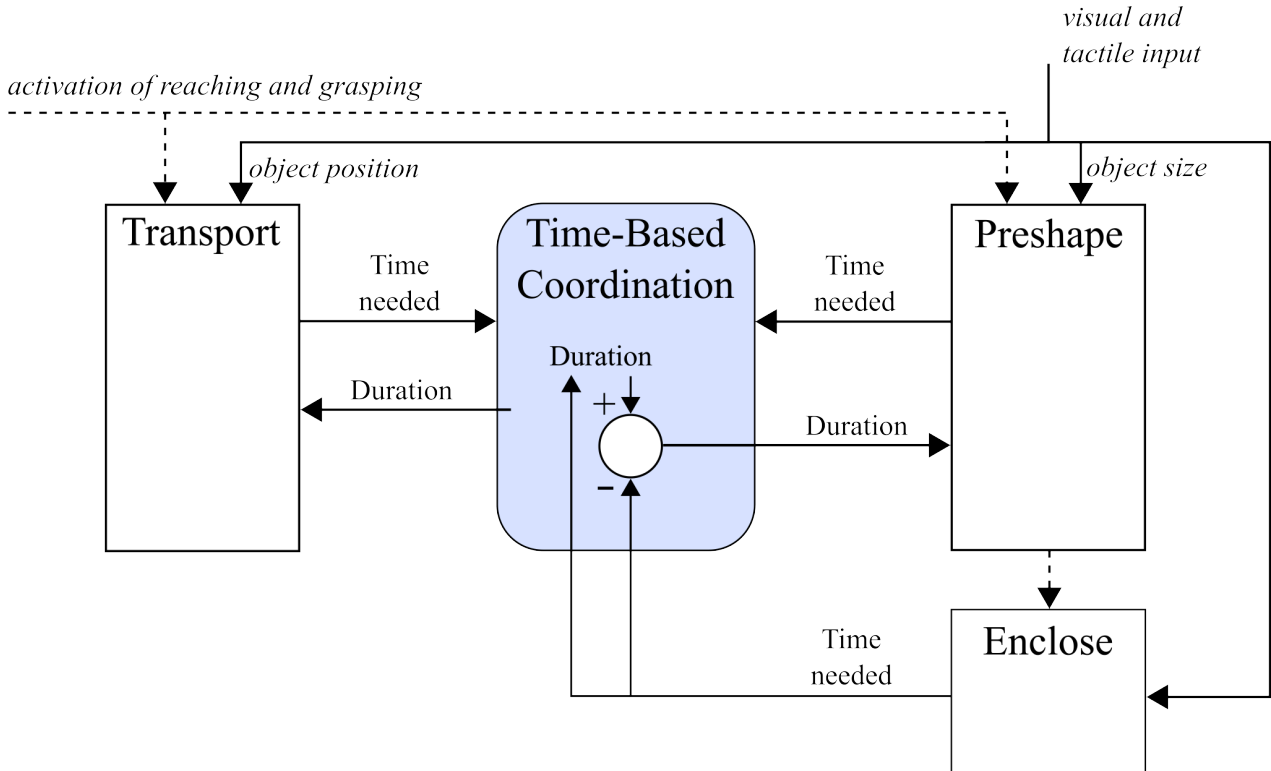


Figure 2.3.: The Hoff-Arbib model as introduced in (Hoff and Arbib, 1993) explaining the temporal coordination and synchronization of the various grasp stages in human grasping, the preshaping and enclosing of the fingers and the approach movement of the hand.

environment, whereas the hand aperture relies on intrinsic properties such as the object size. During the execution of a grasp, transport and grip component are running in parallel. A temporal coupling ensures the synchronization between the components.

Following this idea, (Hoff and Arbib, 1993) introduced a model for grasping, consisting of three independent feedback controllers for approach, preshape, and enclose which are coupled via a time-based coordination module. The controllers are implemented based on a minimum jerk model. In addition, the preshape and enclose controllers comprise an optimization principle which assigns costs for maintaining the hand in an open position as well as for changing the aperture size. The input parameters to the model are the distance to the target object and its size. Based on these parameters the duration of the hand approach movement as well as the maximum hand aperture is determined. Since the enclose time is assumed to be constant for a particular task, the difference between approach duration and the enclose time is considered to be the duration of the preshape phase. The controllers for hand approach and prehensile finger movements are working in parallel and do not share any information about the progress of each other. Therefore, perturbations affecting a single channel are handled by estimating the time needed for the affected controller to terminate. This information is forwarded to the coordination module for the update of the execution times of all controllers. Once the preshape phase is finished, the enclose controller is activated. Following this scheme, the model succeeded in explaining most of the experimental findings regarding the temporal relation between hand approach and aperture in unperturbed grasping as well as grasping under perturbations of object size and positions.

In (Haggard and Wing, 1997), it is argued that the coordination of approach and aperture is based on a rather spatial than temporal relation. This theory is supported by experiments in which subjects are asked to grasp under pull-perturbations affecting the hand transport.

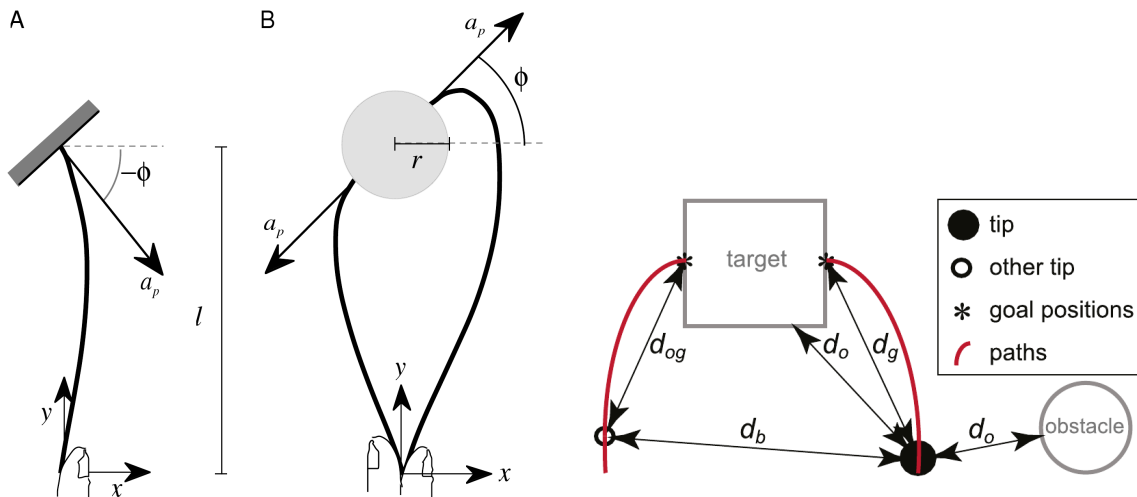


Figure 2.4.: Left: Grasp models for a single finger and for two virtual fingers based on individual digit assumption as proposed in (Smeets and Brenner, 1999). Right: Model of grasp kinematics of two virtual fingertips using spring elements (see (Verheij et al., 2012)).

It was observed that compensation movements, in the form of a coordinated hand aperture response to the perturbation, are performed in order to maintain a spatial relation between the transport and grip component which is featured in unperturbed grasping. This spatial relation is described as a sensorimotor matrix comprising the position and gains. In (Rand et al., 2008), this argument is pursued by assuming the existence of an underlying control law for the initiation of the enclose phase which takes into account the dynamics of the arm when reaching a target. Grasping experiments revealed that the distance of the hand to the target at which the enclose phase starts changed according to the performed approach velocity. This finding indicates the distance to the object at which the enclose phase is initiated depends on the maximum hand aperture, wrist velocity and acceleration. In order to describe the coordination of the hand approach and aperture for the entire grasp process, in (Rand et al., 2008), the control law is extended towards a generalized model which is built upon the relationship between velocity and acceleration parameters featured by the aperture and transport movement. This model features a higher model fitness to experimental grasping data than alternative models which have been proposed so far. Especially, the coordinated hand-approach-enclose movement has been explained more accurately.

A different view on grasping is provided by (Smeets and Brenner, 1999) which is based on the assumption that the kinematics of the fingers is not determined by the hand aperture but by individual finger movements. This assumption is based on the requirement of an oppositional configuration of finger and thumb in order to attain a stable grasp which leads to the hypothesis that not only object size but also the finger contact positions on the object surface are dominating parameters in human grasping. Furthermore, experimental findings indicate that the thumb and virtual finger trajectories are generated independently. These trajectories are modeled using a minimum jerk approach for representing pointing movements. Using this alternative model, several essential grasping characteristics have been verified such as the independence of the transport and prehension component.

Based on the work presented in (Smeets and Brenner, 1999), a dynamical systems model for describing the kinematics of grasping while taking obstacle avoidance as task constraint is proposed in (Verheij et al., 2012). Thumb and index fingertip are modeled as point masses whose movements are determined by a force field consisting of four components. The main force component pulls a single fingertip towards its designated goal, while repulsive forces push

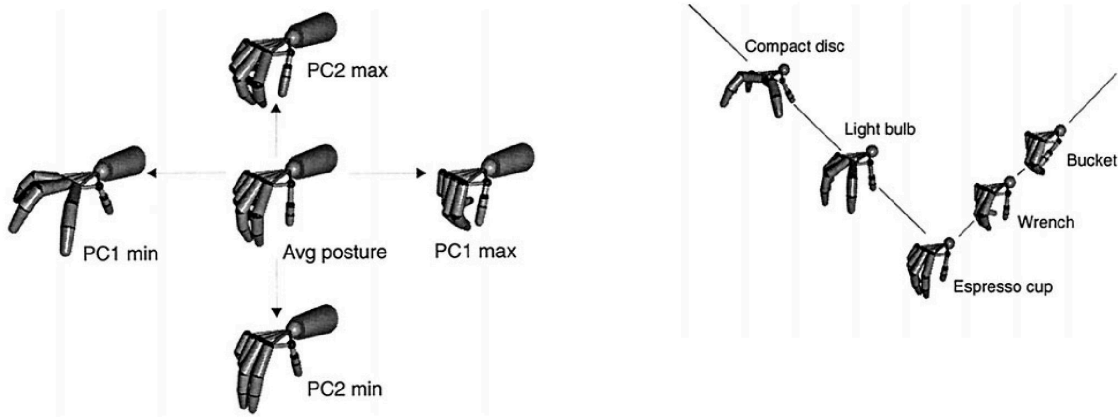


Figure 2.5.: Left: Postural synergies defined by the first two Principal Components as defined in (Santello et al., 1998). Right: Interpolation between various grasp postures described the two PCs showing the grasp synergisties for similar shaped objects. (©1998 by the Society for Neuroscience.)

the fingertips away from any obstacles. A virtual spring between the fingertips generates a synergistic force component in order to avoid collision between the fingers. Simultaneous termination of the movement is accomplished by introducing a damping component to the system. Experimental findings of grasping characteristics as reported in previous studies including grasping under object perturbations has been reproduced using this model. Furthermore, the model is able to deal with online perturbations, meaning perturbations occurring during grasp execution.

### 2.2.1. Postural Hand Synergies

Neuroscientific studies demonstrated that not all finger joints are controlled independently when performing the task of grasping. It is rather suggested that the movements of the finger joints are strongly correlated, and, thus, the grasping movement is dominated by synergies in posture space (see (Santello et al., 1998)). To prove this assumption, experiments showed that static hand postures associated with grasps of familiar objects can be described by a point in low-dimensional latent space. Human subjects were asked to perform grasps for various objects. The hand shaping movement has been discretized in hand postures. Each posture describes a joint angle configuration of the human hand which has been approximated by a model with 15 DoF. The application of the Principal Component Analysis on this data revealed that the variance of the data is large regarding the first components whereas the variance becomes small starting with the fourth component. Subsequently, a low-dimensional representation based on the first three principal components accounts for 97 % of the hand postures whereas with two components 80 % can be represented. The average error when reconstructing a low-dimensional grasp representation is  $\pm 5$  degrees per joint. This implies that only a small number of control variables is needed to actually control a multi-DoF hand when performing grasping actions. Furthermore, a very important conclusion that has been made in (Santello et al., 1998) is that the control of the prehensile finger movements is to be considered separate from the control of the contact forces. This conclusion is based on the observation that although the same hand shapes have been applied to grasp the same object, very different contact forces have been exerted on it.

### 3. Grasp Representations in Robotics

The common methodology to synthesize grasps on robotic platforms consists of a search for an appropriate endeffector pose at which by closing the fingers around the object a grasp is obtained that satisfies constraints imposed by a certain grasp measure. The development of most grasp measures has been mainly driven by grasp stability issues and, hence, can be traced back to the concept of force-closure (Salisbury and Roth, 1983). A grasp meets the force-closure condition when the corresponding contact positions allow a robot to fixate an object regardless of any external forces and moments. Such grasps are represented by a set of contact points and their associated friction cones which describe the location and the direction of the forces which have to be exerted by the fingertips in order to achieve a grasp in the equilibrium, a state at which all forces and moments are balanced out. Based on the determined contacts and the corresponding hand pose, a collision-free grasping movement can be planned and executed. Hence, the most basic grasp representation incorporates the following information:

- **Contact information:** the decision where to place the fingers on the object surface,
- **Movement strategy:** the strategy how the fingers should be moved towards the contact positions.

However, the search for an optimal grasp entails high computational costs even for basic tasks such as pick and place operations with simple robotic platforms. Modern robotics applications comprise manipulation problems which mostly require dexterous object grasping in a task-specific way. Due to the high-level of dexterity featured by the human hand in gross and fine motor skills, in the past years, research efforts have been devoted to development of anthropomorphic robotic hand systems. In this context, various hand designs have been proposed, starting with the Salisbury hand (Salisbury, 1983), the MIT/UTah hand (Jacobsen et al., 1986), the ARMAR-III Hand (Gaiser et al., 2008) and advancing towards sophisticated systems which incorporate a large number of joints and actuators in order to mimic functionality of the human hand. Examples for such systems are the Shadow hand (Shadow Robot

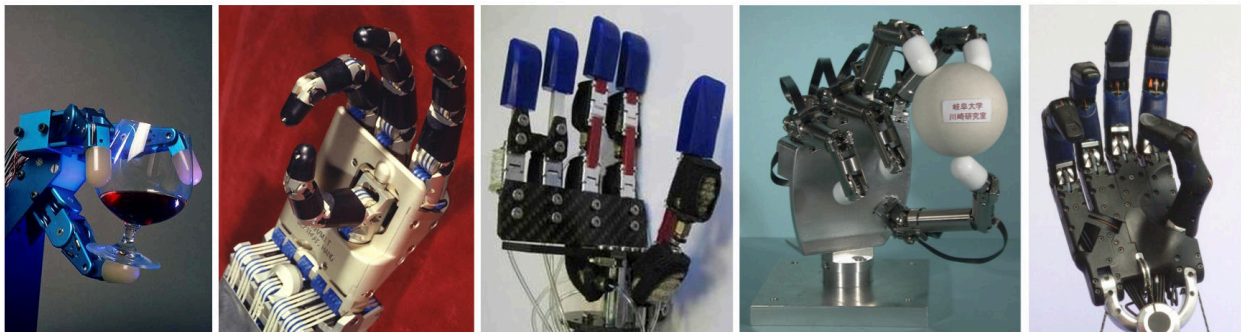


Figure 3.1.: Various robotic hand systems. From left to right: the Salisbury hand (Salisbury, 1983), the MIT/UTah hand (Jacobsen et al., 1986) (©1986 IEEE), the ARMAR-III hand (Schulz et al., 2001) (©2001 IEEE), the Gifu hand (Kawasaki et al., 2002) (©2002 IEEE), and the Shadow hand [Shadow Robot Company] (©2014 Shadow Robot Company).

Company, 2013) with 24 DoF and the Gifu hand (Kawasaki et al., 2002) with 22 DoF which are capable of performing highly dexterous manipulation tasks. However, the dexterity comes with an increase in size, weight, and most importantly control complexity. As a consequence, the search space increases dramatically with the complexity as well as the additional task-specific constraints that have to be determined and considered. This makes the computation of a grasping action with classical grasp planning methods intractable. With regard to the anthropomorphic design of robotic platforms, an intuitive approach is to derive heuristics from the observation of human grasp behaviors. In order to integrate this knowledge, grasp representations have to be extended and designed such that knowledge about human grasping can be encoded in a form that is accessible to robots. In the following, grasp representations are discussed which are inspired by human grasping and are used for the recognition, analysis, and synthesis of grasps in robotics applications. In the perspective of this thesis, only approaches which allow the representation of prehensile finger movements are reviewed.

### 3.1. Human-Inspired Discrete Grasp Representations

A straightforward approach to reduce the complexity in robotic grasp synthesis is to use human grasp taxonomies, as presented in Section 2.1, in order to constrain the grasp configuration space. In doing so, the discrete set of commonly applied grasp postures can be exploited as pregrasp or initial solutions which serve as starting points to the search for a suitable grasp pose which satisfies the given constraints. In this context, motivated by (Iberall, 1997), a grasp representation is proposed in (Rao et al., 1989) which is based on eight different generalized grasp types: power grasp, span grasp, cylindrical grasp, chuck grip, hook grip, lateral pinch, pulp pinch, and precision pinch. Each grasp type determines a joint angle range for each motor of a finger. For the grasp synthesis, the object of interest is localized in the scene and

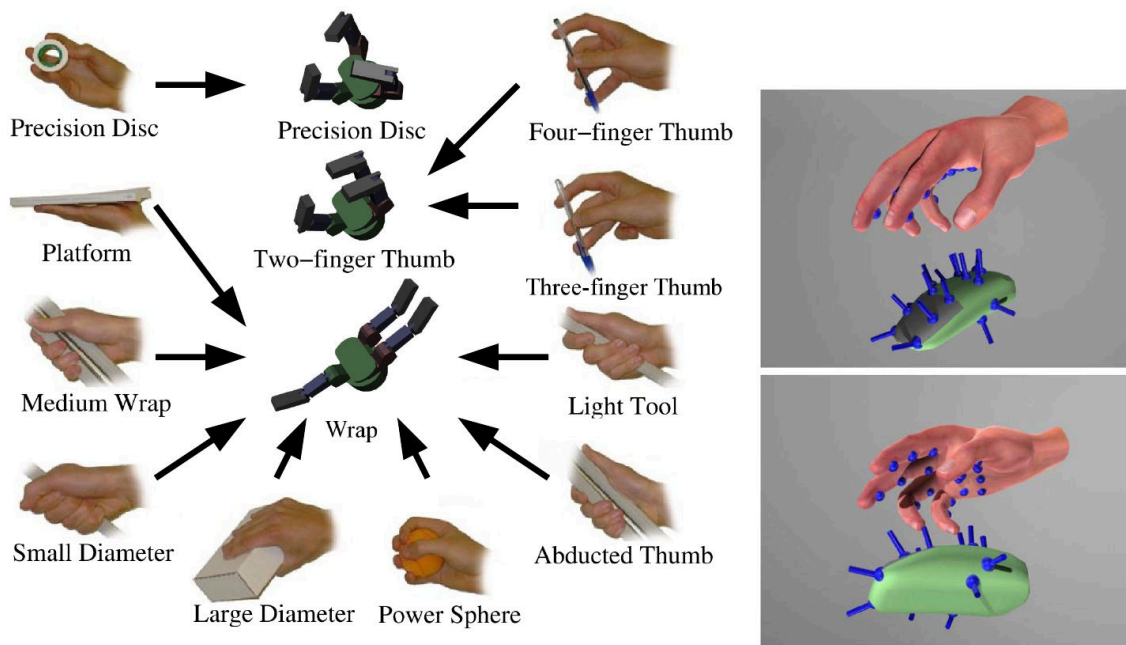


Figure 3.2.: Left: Mapping a selection of grasps from Cutkosky grasp taxonomy to the Barrett hand (Tegin et al., 2009). (©2009 IEEE.) Right: Grasp synthesis as a shape matching problem (Li and Pollard, 2005). A hand shape represented by contact points and normals is determined which provides the best match with regard to 3D object model. Here, a grasp posture for a mouse is depicted. (©2005 IEEE.)

represented in the form of geometric primitives (e.g. boxes, spheres, cylinders, and cones) whereby all grasp types are adapted to each primitive. Depending on given task constraints, the most qualified grasp type is executed.

In (Tegin et al., 2009), a programming by demonstration framework for grasping is presented in which demonstrated human grasps are classified based on the Cutkosky taxonomy and related to a specific object and a corresponding approach vector. As depicted in Figure 3.2, a grasp is represented by a robot hand configuration which results from a learned mapping of human grasp types to the kinematic structure of the robot hand. For the grasp reproduction, based on human grasp demonstrations, the grasp type is recognized. The corresponding prototypical hand configuration is aligned with the pose of the object to be grasped and, thus, forms a pregrasp posture which serves as an initial solution to a grasp planning algorithm.

In (Li and Pollard, 2005), a grasp synthesis approach based on a shape matching algorithm is developed. As depicted in Figure 3.2, a hand shape is represented in the form of contact points and the corresponding contact normals for an object which is described by a dense cloud of oriented points. Based on a database containing object- and task-specific grasp postures recorded from human demonstration, a selection of possible grasp posture hypotheses is retrieved by matching object and hand shape features. Using a clustering algorithm and task constraint pruning, the most suitable grasping hand shape is determined and additionally refined to obtain a stable grasp.

In (Kjellstrom et al., 2008), a framework is proposed for the recognition and the reproduction of grasps demonstrated by a human. Based on a database which comprises a large number of different views of a human hand performing different grasp types from various






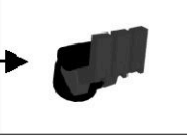



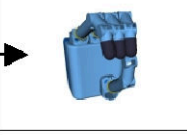






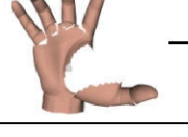

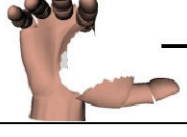

Eigengrasp 1			Eigengrasp 2		
Description	min	max	Description	min	max
Prox. joints flexion			Dist. joints flexion		
Spread angle opening			Finger flexion		
Prox. joints flexion Finger abduction			Dist. joints flexion Thumb flexion		
Thumb flexion MCP flexion Index abduction			Thumb flexion MCP extension PIP flexion		
Thumb rotation Thumb flexion MCP flexion Index abduction			Thumb flexion MCP extension PIP flexion		

Figure 3.3.: Illustration of first two Eigengrasps for several hand embodiments according to (Ciocarlie and Allen, 2009). For each hand, these Eigengrasps are extracted from training data which contains numerous grasp postures mapped to the kinematics of the hand. (©2006 SAGE Publications.)

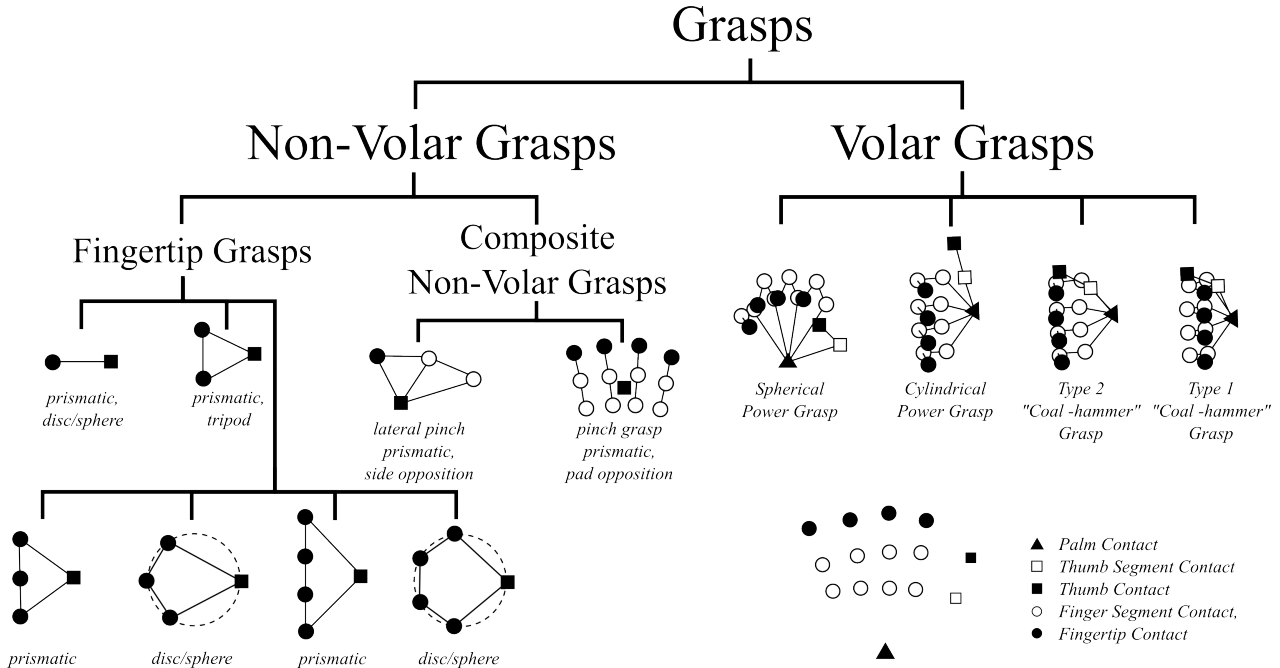


Figure 3.4.: Based on the Contact Web representation, in (Kang and Ikeuchi, 1997), a grasp taxonomy is introduced which mainly distinguishes between volar and non-volar grasps. grasp configurations representing different grasp types are visualized as graph structures whose vertices are depicted as circles and squares. The bottom right corner shows the key to the used symbols as well as the right human hand in opened configuration using the Contact Web. (©1997 IEEE.)

grasp taxonomies (see (Cutkosky, 1989), (Kang and Ikeuchi, 1997), and (Kamakura, 1980)), the robot is enabled to track and observe a grasp applied on a specific object. The view obtained by observation of the demonstration is matched with hand shapes in the database to obtain the applied grasp type and the corresponding approach vector. For the reproduction, each recognized grasp type is associated with a prototypical joint angle configuration and the corresponding fixed grip aperture. To adapt this prototype to different situations, the size of the object is visually determined and the grip aperture is adjusted accordingly.

Motivated by (Santello et al., 1998), the concept of the Eigengrasps has been introduced concept in (Ciocarlie and Allen, 2009) in order to attain a tradeoff between complexity and controllability. As depicted in Figure 3.3, the Eigengrasp is determined by deriving a projection function which allows the description of a multi-DoF robot hand configuration in the form of a low-dimensional vector in latent space. For this purpose, the Principal Component Analysis is applied on a postural space containing common human grasp postures which are mapped to a specific robotic hand. As indicated in (Santello et al., 1998), a grasp posture can be represented by a linear combination in the Eigengrasp space. This space is spanned by the first two eigenvectors of the data covariance matrix which correspond to the two largest eigenvalues. For the synthesis of a grasp, a suitable posture for a given task is selected and optimized with regard to a grasp quality function. Due to the low-dimensional representation, a grasping action can be planned and evaluated in an efficient manner. However, this approach does not scale to different objects as well as to other robotic hands. Consequently, individual Eigengrasp spaces have to be generated for each embodiment.

In (Kang and Ikeuchi, 1997), a grasp is represented by the Contact Web which is a graph structure in 3D whose nodes describe contact points between the hand and the object. As depicted in Figure 3.4, each segment of the human hand features a contact point, hence, a



grasp is characterized by the shape and cardinality of the graph. For the recognition and analysis of a grasp, based on the Contact Web representation, a taxonomy is derived which distinguishes between volar and non-volar grasps depending on whether a palm contact exists or not. In order to map a specific grasp type to a robotic hand, the nodes of the corresponding Contact Web are grouped to virtual fingers. Each virtual finger is manually associated to an actual robotic finger. For the synthesis of the mapped grasp, the Contact Web is parameterized based on the contact points and the hand approach information extracted from human grasp examples.

### 3.2. Human-Inspired Continuous Grasp Representations

The representations discussed in Section 3.1 provide a rather static description of grasping actions which provides a search space reduction and, thus, leads to an improvement of the grasp planning process. However, they lack essential information based on which a movement strategy for prehensile finger movements can be inferred. In humans, hand approach and prehensile finger movements are intertwined allowing the immediate adaptation of a grasping strategy in order to accommodate changes in the scene and object properties. Hence, to implement an efficient grasping behavior in dynamic environments, the question of how finger movements can be incorporated in a grasp representation has to be addressed. Therefore, instead of reducing a grasp to a single hand posture which is a mere snapshot of this continuous process, a grasp ought to be represented in a continuous fashion as well.

In the following, grasp representations which follow a continuous approach, comprising a motion description of the finger trajectories from preshape, approach, and enclose, are reviewed. Approaches can be coarsely subdivided based on the configuration space in which the finger movements are defined: continuous representations in the joint angle space and in the task space.

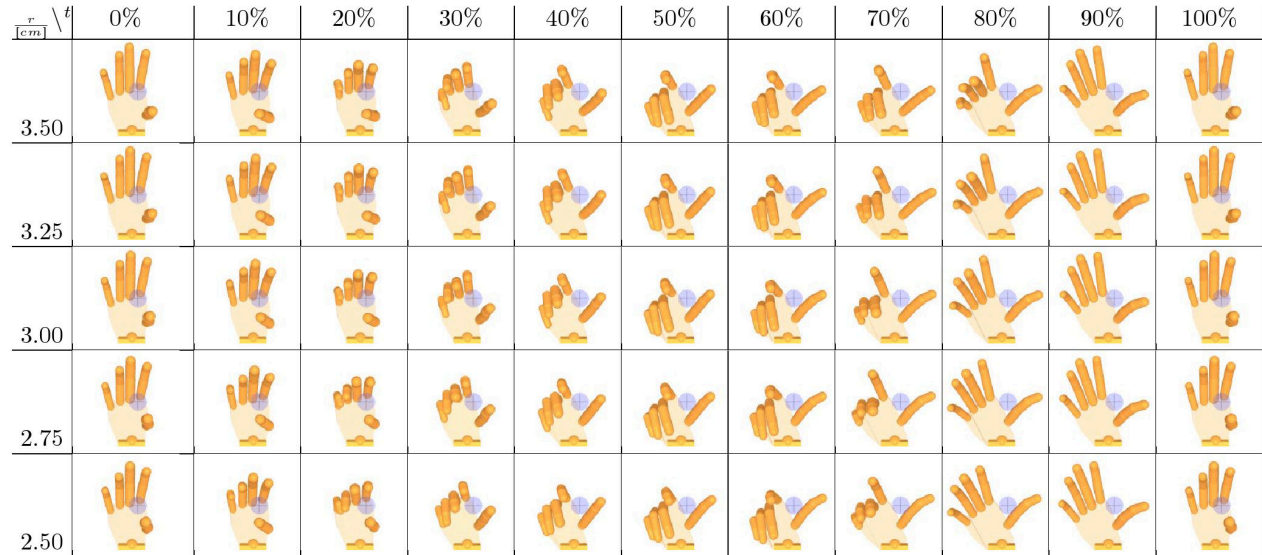


Figure 3.5.: Manipulation manifolds for a cap turning movement as introduced in (Steffen et al., 2008). The manifolds are arranged vertically according to the radius of the cap and horizontally according to the temporal evolution. (©2008 IEEE.)

### 3.2.1. Representations in Joint Space

An intuitive approach to create a continuous grasping movement is the interpolation between grasp postures which are temporally related. Following this idea, (Nguyen and Stephanou, 1990) suggest a mapping from the hand configuration space to a topological space which is defined by four terminal hand configurations (full adduction, abduction, fully flexed, fingertips touched). Thus, arbitrary hand postures are approximated by a linear combination of these four hand configurations. To obtain a continuous grasping movement, points representing the preshape and final grasp posture which satisfy object- and task-specific constraints have to be determined. The interpolation between these two postures results in a continuous grasping movement.

A more elaborate approach in the form of manipulation manifolds has been proposed in (Steffen et al., 2008). The core component of a manipulation manifold consists of a continuous mapping which relates an ordered sequence of hand configurations to a series of latent lower dimensional feature vectors. In order to obtain an approximation of this mapping, unsupervised kernel regression is applied on human grasp data. As depicted in Figure 3.5, manipulation manifolds have been used for the representation of a cap turning movement. Subsequently, the latent feature vector consists of a normalized time stamp and the diameter of the cap which relates the encoded movement to object-specific properties. Perturbations in object size require training of a novel manifold. All manifolds constitute a map which represents a specific movement generalized to different intrinsic object properties. Using these manipulation manifolds, a grasp can be represented in a similar manner. However, the main drawback of this approach lies in the scalability of manifolds. To cover a wide range of grasping movements for different objects, a large amount of grasping data has to be collected and considered in the model building process.

An alternative way to derive a mapping from object properties to grasp configurations is to use neural networks. Following this methodology, in (Uno et al., 1995) an approach is proposed to generate a model with which a grasp hand posture is inferred based on a visual image of the object to be grasped. Without explicitly specifying the relations between the object and the hand's kinematics, this model is derived by training a neural network with data from the observation of human grasps which consists of lateral object views annotated with

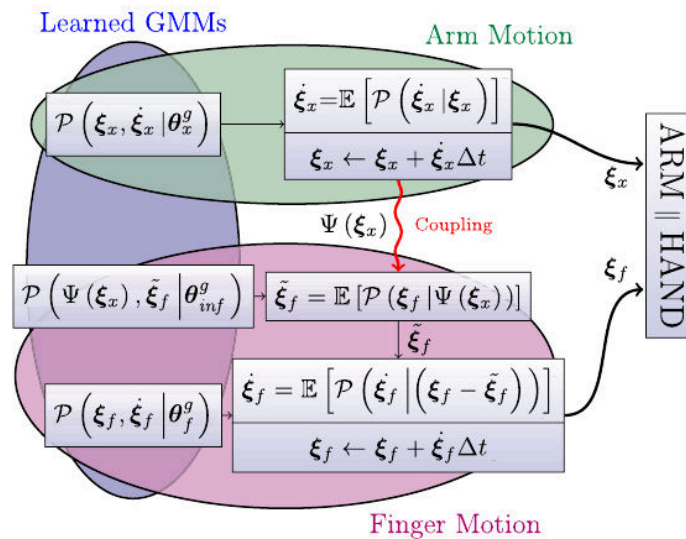


Figure 3.6.: Representation of a grasping primitive using the coupled dynamical systems approach as presented in (Shukla and Billard, 2012). (©2012 Elsevier B.V.)

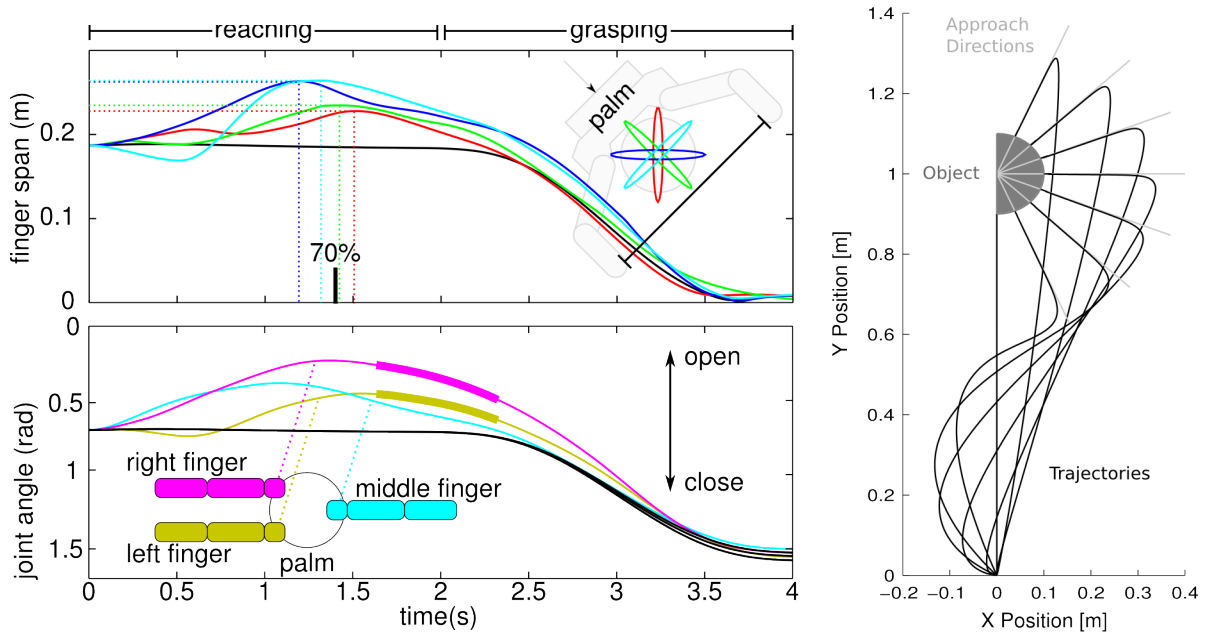


Figure 3.7.: Left: Continuous representation of prehensile finger movements defined in joint space using the DMP representation as introduced (Stulp et al., 2011). Right: Reproduction of fingertip trajectories in task space represented in the form of a DMP as defined in (Kroemer et al., 2010). (©2010 2011 IEEE.)

the corresponding grasp hand posture. In (Taha et al., 1997), an approach is presented which enables the reproduction of the entire grasping sequence, from preshaping to enclosing. To do so, a neural network is trained with continuous grasp data represented as temporal sequences of joint angle movements related to specific objects. Experiments showed that the resulting representations are very limited in their capabilities to adapt to other objects, different tasks or embodiments.

In (Jäkel et al., 2010), grasping strategies are encoded in a representation which combines the wrist approach movement in task space and the prehensile finger movements in joint angle space. Embedded in a programming by demonstration framework, human grasp demonstrations are generalized to strategies which can also be applied to grasp slightly different objects. This is accomplished by incorporating a probabilistic variation model which combines various error models regarding the object localization and the mapping. This model induces a non-uniform sampling distribution of the configuration space and is used to guide the search of a probabilistic motion planner.

In (Shukla and Billard, 2012), a grasp controller is implemented based on coupled dynamical systems which considers the hand as well as the finger movements. Both movements are described by separate dynamical systems which use ordinary first order differential equations for the encoding of postural endeffector movements. To determine these differential equations based on demonstrations of a specific grasp, it is assumed that each point in the demonstrations is drawn from a joint distribution and that the probability density of the data is modeled as a Gaussian Mixture Model (GMM). Both dynamical systems, represented as separate GMMs, are coupled via a third GMM which encodes the joint probability distribution of the inferred state of the fingers and the current hand pose. The proposed approach is capable of encoding a specific grasping movement which is adapted to a specific object. However, it does not generalize well to different task- and object-specific constraints and, thus, has to be learned for every object and grasp type.

A similar approach is introduced in (Stulp et al., 2011), where a Dynamic Movement Primi-

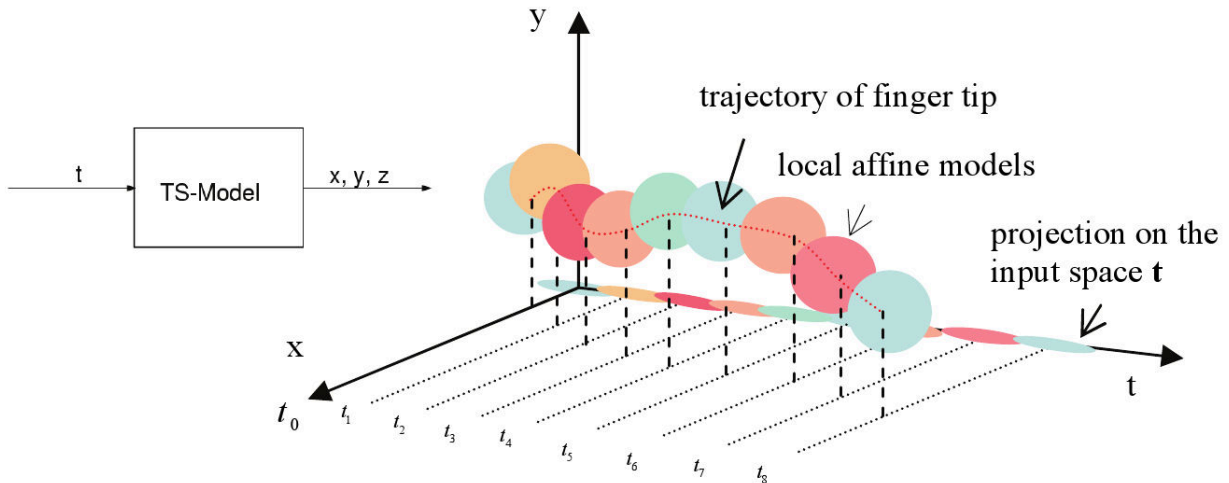


Figure 3.8.: Representation of the fingertip movements in the form of clusters projected into low-dimensional latent space (Palm and Iliev, 2006). (©2006 IEEE.)

tive (DMP) is used to encode the coupled hand and finger movements. A DMP is a movement representation which consists of a set of differential equations each describing a mass spring damper system attached to a goal position and perturbed by a weighted nonlinear force term. Each dynamical system, denoted as transformation system, represents a single dimension of the movement to be encoded. An additional dynamical system, the canonical system, is imposed on the transformation systems in order to control their temporal evolution. For an elaborate discussion of the DMP concept the reader is referred to Section 4.4.1. A grasping movement is described by a  $(6 + N)$ -dimensional trajectory which represents the movement of the wrist and  $N$  finger joints. Based on a simulated grasping movement starting from a preshape posture and ending at the final grasp posture, a DMP is initially learned. Using a reinforcement learning approach, the DMP is refined in order to generalize the grasp representation to varying object positions.

An approach which also uses DMPs to describe dynamics of grasping movements in latent space is introduced in (Amor et al., 2012). Similar to the Eigengrasp approach a low-dimensional subspace is inferred by applying PCA on a set of human grasp demonstrations which reflect different grasp types. By extracting the first five principal components a mapping between the hand joint angle space and a low-dimensional subspace is established. In order to learn a specific grasp type, a corresponding grasping movement is projected into this subspace. Based on the projection a DMP is learned where each dynamical system represents a single dimension of the latent space.

### 3.2.2. Representations in Task Space

In (Palm and Iliev, 2006), for the purpose of recognizing human grasps, fingertip trajectories are modeled as a set of clusters which are formed at discrete times. A time-dependent weighting function connects the clusters to create a continuous description of the fingertip trajectories in task space. Using a set of fuzzy rules and predefined norm with which the similarity to various grasp prototypes is determined, an observed grasp sequence is reconstructed and classified.

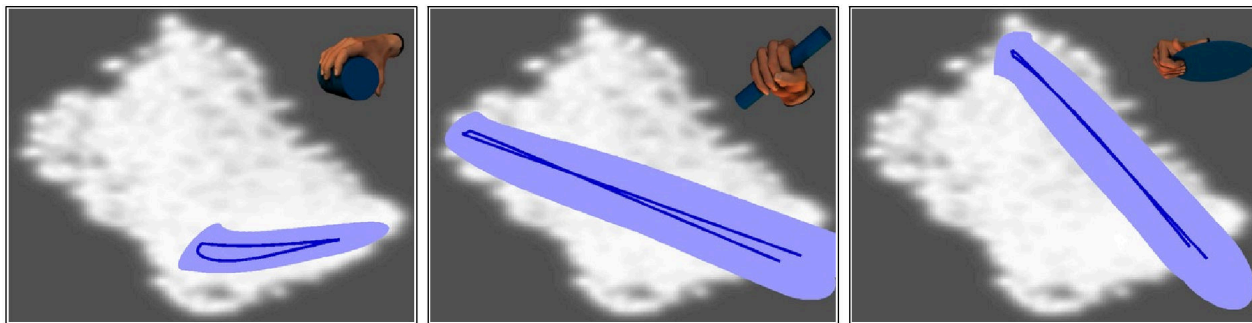


Figure 3.9.: Continuous grasp representation for various grasp types using GMM in low-dimensional latent space as described in (Romero et al., 2010). (©2010 IEEE.)

A low-dimensional approach which exploits synergies in Cartesian space is presented in (Romero et al., 2010). For a compact representation of a grasping movement, a method based on Gaussian Process Latent Models (GPLVM) is applied to create a mapping from the original observational space in which each point represents the fingertip positions as well as the hand pose to a low-dimensional latent space. The use of Gaussian processes to regress the mapping function allows a more accurate reconstruction and the processing of complex and large datasets. Based on different grasp examples, various movements are encoded within the GPLVM subspace. To form a continuous representation, each point in latent space is enriched with a temporal dimension and Gaussian Mixture Models fitted to the trajectory in this space. For the reconstruction of an encoded grasping movement, hand configurations for each time step are inferred using Gaussian Mixture Regression.

In (Kroemer et al., 2010), a continuous grasp representation is presented consisting of a DMP framework where each fingertip movement in task space is encoded as a single DMP which is learned from a grasp example featuring a specific grasp. To synthesize an encoded grasp, the DMPs are parameterized with appropriate contact positions as well as eventual obstacle positions which are represented by a repelling force potential. These parameters are extracted from visual features in the form of 3D edges which can either represent parts of the target object or obstacles within the scene.

### 3.3. Discussion

Representations in joint space have the advantage that a kinematic solution for the control of the fingers is immediately available. However, it is not guaranteed that the hand and the fingers are aligned with the presumed contact positions. In the context of dexterous grasping, hand postures provided by these representation can be merely considered as initial solutions to grasp planning algorithms. Therefore, representations in joint space are limited in their capability to generalize situation-specific grasp observations. In most cases, variations regarding task- and object-specific constraints require extensive training of multiple models in order to implement an efficient grasping behavior. Especially for high-dimensional data which originate from the observation of complex systems such as the human hand, the processing of the grasp data becomes cumbersome.

Representations defined in task space such as the Contact Web introduced in (Kang and Ikeuchi, 1997) allow the direct incorporation of grasp-relevant information about the relationship between hand and object, and, thus, facilitate the transfer of motor knowledge between different embodiments as well as the adaptation to changing object- and task-specific constraints. In order to obtain a comprehensive grasp representation in the task space, movements of the hand as well as the fingers have to be encoded in a continuous fashion. Few works such

as (Romero et al., 2010) and (Kroemer et al., 2010) addressed this issue where different assumptions have been made. In (Romero et al., 2010), the fingertip movements are considered as a trajectory of a subsuming entity. This approach is robust to variances in grasp examples performed by different human subjects and facilitates a low-dimensional grasp representation as well as the incorporation of finger movement synergies. However, focusing on grasp classification, the approach studies how different grasp performances can be structured rather than addressing the question of how a grasp can be reproduced under different conditions. As opposed to this work, (Kroemer et al., 2010) developed a representation for the generation of grasp primitives that can be reproduced with changing object- and task requirements. This approach is based on the assumption that each finger moves individually, and, thus, disregards kinematic dependencies and constraints as well as finger movement synergies which might lead to infeasible solutions.

In this thesis, a continuous grasp representation is proposed which allows the continuous description of a grasping process observed in task space. In the context of grasp learning from human observation, finger movement synergies are modeled and integrated in this representation in order to form an adequate construct based on which robotic grasp primitives can be derived in an efficient manner.

## 4. Grasp Representation

The variety of numerous grasping possibilities and the continuous emergence of novel situations in human-centered environments, requires the ability for robots to rapidly acquire grasp-related motor knowledge. A methodology which supports the development of such an efficient grasp learning behavior is learning from observation. Based on observations of a specific grasping movement, a representation is desired which, on the one hand, encodes relevant movement characteristics for the observed grasp type and, on the other hand, generalizes the observed movement from the specific context. A grasp representation should contain sufficient information allowing to infer a grasping movement policy which yields a similar behavior as featured in the observed grasp demonstrations and which can be adapted to current object and task-specific constraints. In this chapter, a continuous grasp representation in the task space is proposed with required attributes in order to implement such grasp learning behavior. The representation incorporates two separate motion models for the description of the prehensile finger and hand movements. Especially, the question of how prehensile finger movements can be represented is addressed in this chapter. For this purpose, a novel approach in the form of the Virtual Spring Grasp representation is proposed. First, in Section 4.1, this approach for a continuous grasp representation is motivated. In Section 4.2, fundamental principles of the representation are described. Subsequently, the two motion models are elucidated. Section 4.3 introduces the representation for the modeling of finger movements. The motion model for the representation of hand approach movement is described in Section 4.4. The coupling between both is addressed in Section 4.5.

### 4.1. Motivation

A continuous approach bears the advantage that it facilitates the adaptation of the representation to varying constraints and, thus, allowing to attain goal-directed grasping movements in dynamic scenes. As suggested by (Hoff and Arbib, 1993), the proposed grasp representation comprises two separate models, one for the representation of the hand movement and one for the finger movements. In the following, a grasping movement performed by a hand with  $N$  fingers is described by a trajectory  $\mathbf{G} := (\mathbf{X}_H, \mathbf{X}_F)$  consisting of  $T$  frames where the hand approach movement is represented by the trajectory matrix  $\mathbf{X}_H \in \mathbb{R}^{6 \times T}$ :

$$\mathbf{X}_H = (\mathbf{x}_H(1) \dots \mathbf{x}_H(T)) \quad (4.1)$$

with  $\mathbf{x}_H(t) = (x_H(t), y_H(t), z_H(t), \alpha_H(t), \beta_H(t), \gamma_H(t))^T$  denoting the palm position and orientation at time  $1 \leq t \leq T$ . The finger movements are described by the trajectory matrix  $\mathbf{X}_F \in \mathbb{R}^{3 \cdot N \times T}$ :

$$\mathbf{X}_F = \left( \left( \begin{pmatrix} \mathbf{x}_1(1) \\ \vdots \\ \mathbf{x}_N(1) \end{pmatrix} \dots \begin{pmatrix} \mathbf{x}_1(T) \\ \vdots \\ \mathbf{x}_N(T) \end{pmatrix} \right) \right) \quad (4.2)$$

with  $\mathbf{x}_i(t) = (x_i(t), y_i(t), z_i(t))^T$  indicating the position of fingertip  $i$  with  $1 \leq i \leq N$  which is defined in a coordinate system located within the hand. To obtain a continuous representation of a grasping movement, a model which approximates  $\mathbf{G}$  is needed. A method to approach this problem is the use of dynamical systems which can be applied to describe the temporal

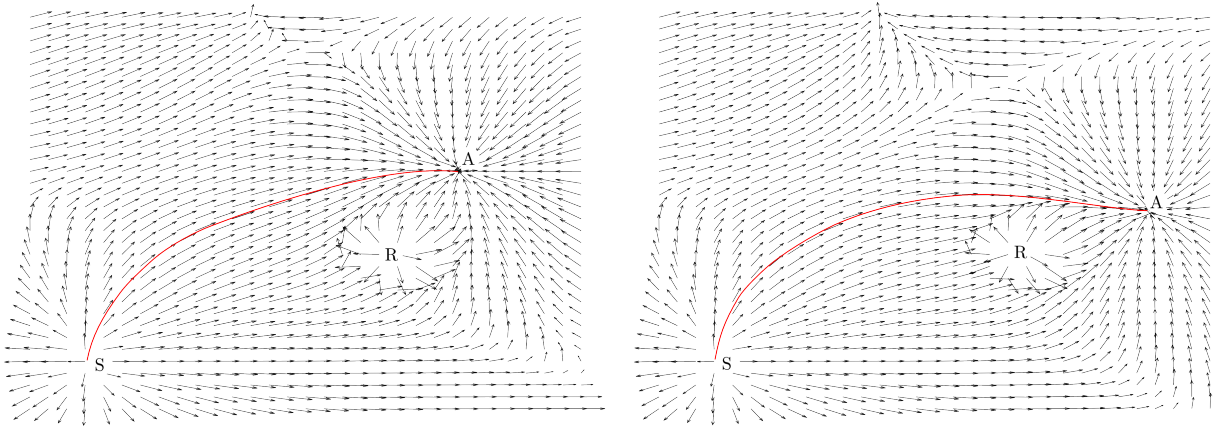


Figure 4.1.: Left: 2D vector field which drives a mass body from a start position  $S = (1, 1)$  to a goal position  $A = (7, 6)$ . A repelling force emanates from  $R = (6, 4)$ . Right: Same vector field as depicted in the left figure with a shifted goal position  $A = (8, 5)$ .

evolution of an underlying physical system. In terms of grasping, the physical system is model of the grasping hand. A state of a dynamical system imposed on this model corresponds to a certain hand configuration. Movement representations based on dynamical systems have been proposed in various works (see (Ijspeert et al., 2002), (Schaal et al., 2005), (Hersch et al., 2008), (Ude et al., 2010)). Based on a formalization using differential equations, as illustrated in Figure 4.1, an attractor landscape is shaped which determines how the physical system evolves over time and which defines a specific goal state to which the system converges. In a similar fashion, a repeller can be used to prevent the system of adopting an infeasible state. Especially, considering goal-directed movements, these properties allow the generation of movement policies which can be easily adapted to different start and goal conditions and are robust against perturbations allowing to efficiently accommodate dynamic changes in the environment.

In the following, the capability of dynamical systems to represent coordinated movements performed by multiple fingers is investigated. Previous approaches for the representation of finger movements are defined in a common space in which each point represents a configuration for all fingers and, thus, treating fingers as a single entity. This methodology facilitates the description of coordinated finger behavior, but, however, does not allow the consideration of individual finger movements. This property can be beneficial and under certain conditions is necessary to generate goal-directed grasping movements.

To enhance the adaptivity and flexibility of a grasp representation, the presented approach is based on the assumption raised by (Smeets and Brenner, 1999) which suggests that pre-shaping and enclosing of the fingers can be described as synchronized movements of individual digits in the task space. This view on grasping allows the direct consideration of external disturbances affecting an individual finger and, therefore, enables more accurate responses to perturbations in object position and size. Hence, inspired by these findings and based on the Contact Web representation introduced in (Kang and Ikeuchi, 1997), the Virtual Spring Grasp Representation is introduced which interprets each finger and contact point as a mass body in the task space. The synergistic effects between the bodies are implemented using virtual spring components. For the representation of the hand approach movement, the proposed approach relies on Dynamic Movement Primitives, a movement representation based on nonlinear differential equations. Both representations are joined via a canonical system in order to establish a coupling between the hand approach and the finger movements.



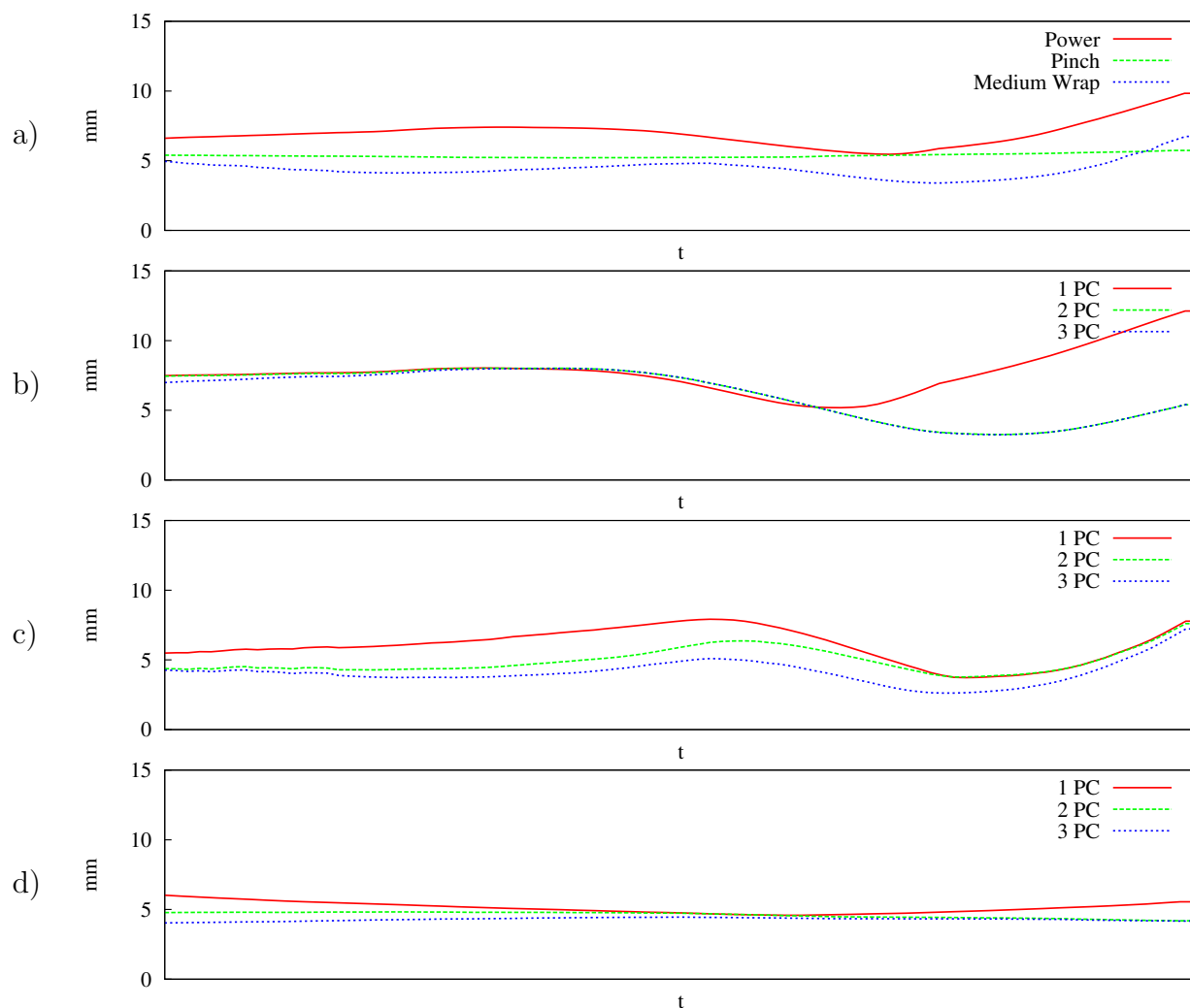


Figure 4.2.: a) Backprojection error for grasp examples featuring a power grasp, pinch grasp, and a medium wrap grasp. The 2D latent space is spanned by the first two principal components which are extracted from the entire grasp data. The backprojection error for power grasp examples where based on power grasp data featuring a latent space is derived with 1, 2, or 3 principal components in b). For a medium wrap and for a pinch grasp, the error is plotted in c) respectively d).

#### 4.1.1. Finger Movement Synergies

An important aspect when developing an approach for the representation for the prehensile movements is the consideration of postural finger movement synergies. As indicated by (Santello et al., 1998), synergies are mechanisms which enable the human brain to facilitate the control of the grasping hand. In robotics applications, this concept has been successfully implemented in the form of underactuated hands (see (Brown and Asada, 2007),(Catalano et al., 2012),(Chen and Xiong, 2013)). Such systems are capable of performing dexterous grasping actions with a reduced control complexity due to the low-dimensional control input.

Finger movement synergies have been mostly studied in hand configuration space. For the incorporation of these mechanisms in the proposed representation, the emergence of synergistic effects in the task space is studied. To show that postural synergies lead to synergies in the task space, human grasp demonstrations of different grasp types (power grasp, pinch grasp, lateral grasp, spherical pinch grasp) have been recorded in the form of fingertip trajectories. As suggested by (Santello et al., 1998), by applying a Principal Component Analysis (PCA)

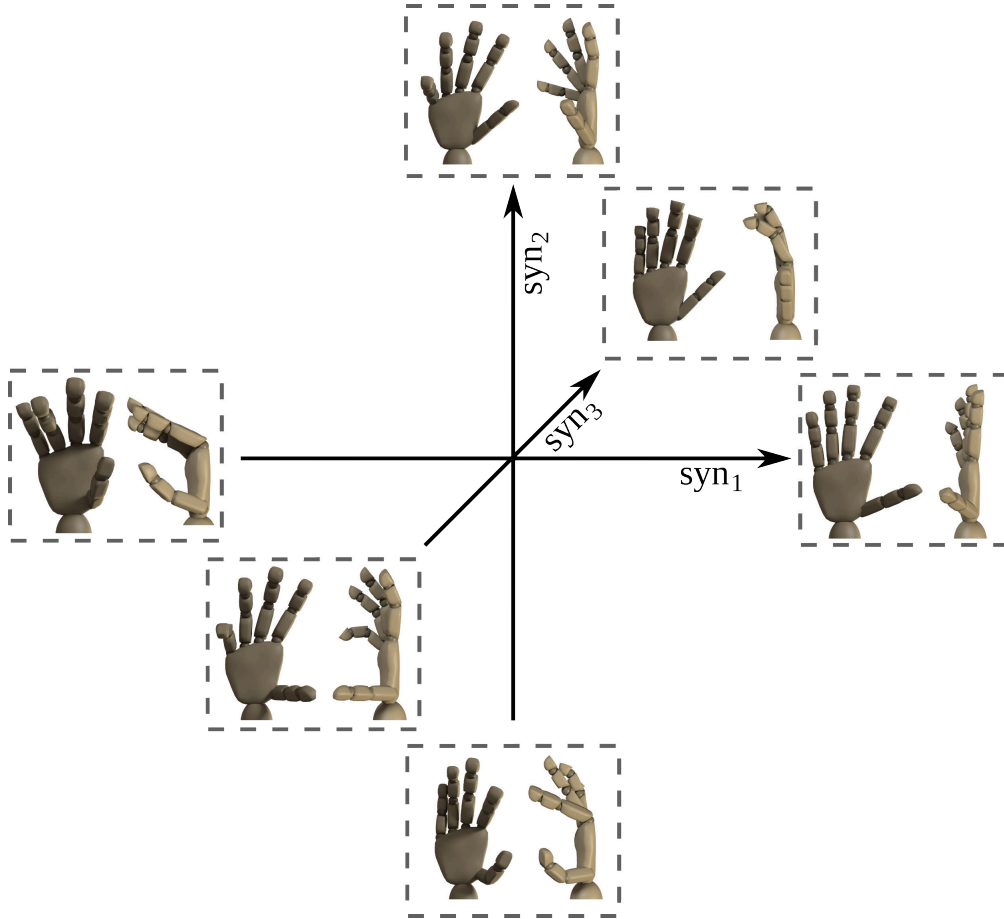


Figure 4.3.: Finger movement synergies in the task space represented by the first three principal components extracted from the grasp demonstrations. The corresponding hand postures depict the minimum (start of an arrow) and the maximum (end of an arrow) values of the principal components.

on the grasp trajectories, a low-dimensional latent space is defined by the first two principal components. For a five-fingered hand, each point of a novel grasp trajectory  $\mathbf{x}_h \in \mathbb{R}^{5 \cdot 3}$  is projected on a latent representation  $\mathbf{x}_l \in \mathbb{R}^2$ . Through backprojection of  $\mathbf{x}_l$  to a fingertip configuration  $\mathbf{x}_h' \in \mathbb{R}^{5 \cdot 3}$ , one can assess how well a fingertip configuration can be represented in latent space by evaluating the difference between  $\mathbf{x}_h'$  and  $\mathbf{x}_h$ . The plot a) in Figure 4.2 depicts the average reconstruction error of a fingertip position which is approximately 5–7 mm depending on the number of principal components used for the projection. In plot b), c), and d), the reconstruction error is visualized for a latent space generated from data which features a specific grasp type. With three principal components the error is reduced to  $\approx 4$  mm. Similar to the findings stated in (Bicchi et al., 2011), the inclusion of a fourth principal component does not considerably improve the experimental results. The latent space spanned by the first three principal components is visualized in Figure 4.3. The finger movement synergies represented by the principal components are denoted as  $\text{syn}_1$ ,  $\text{syn}_2$ , and  $\text{syn}_3$ . Synergy  $\text{syn}_1$  mainly controls the fingers to attain an oppositional configuration between thumb and the remaining fingers while extending the fingers. Synergy  $\text{syn}_2$  addresses the opposition between thumb and the index finger whereas synergy  $\text{syn}_3$  appears to be crucial for the preshaping of the hand, since it represents a hand opening movement by maximizing the distance between the virtual finger and the thumb. The results of this analysis reveal that finger movements do not only feature synergies in joint angle, but in task space as well. The synergies further show that the thumb movement correlates the least with the movements of the other fingers.

This indicates that the thumb plays a particular and decoupled role in grasping and, thus, has to be treated differently from the other fingers.

## 4.2. Basic Principles

The VSG representation is based on the assumption that a fingertip can be represented as a mass body  $b$  in Cartesian space whose position, velocity, and acceleration are denoted as  $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}} \in \mathbb{R}^3$ . For the sake of simplicity, it is assumed that the mass of  $b$  is  $m = 1$ . According to Newton's Second Law the body's acceleration is equivalent to the net forces acting on the body:

$$\mathbf{f} = m\ddot{\mathbf{x}}. \quad (4.3)$$

To constrain the movement of  $b$ , as depicted in Figure 4.5, a mass spring damper system is employed which connects  $b$  to a fixed point in space. For the sake of simplicity, this fixed point is assumed to be the origin of the coordinate system in which the spatial properties of  $b$  are defined. Hence, for a spring with an equilibrium length  $l$  and a spring constant  $k > 0$ , the forces acting on  $b$  in order to restore the equilibrium state are described by following equation:

$$\mathbf{f}_b = -k(d-l)\hat{\mathbf{x}} - \zeta\dot{\mathbf{x}} + \mathbf{f}_e \quad (4.4)$$

with  $\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$  being the unit vector of  $\mathbf{x}$  and  $d = \|\mathbf{x}\|$  describing the distance of  $b$  to the origin. Potential external forces acting on  $b$  are incorporated in  $\mathbf{f}_e$ . With the damping coefficient  $\zeta$  the amplitude of the oscillations after perturbations is controlled. According to (Komkov, 1972), to induce a specific oscillatory behavior into the system,  $\zeta$  has to be determined based on the corresponding damping ratio  $r$ . This is described by following equation:

$$r = \frac{\zeta}{2\sqrt{k}}. \quad (4.5)$$

In general, we distinguish between four different oscillatory behaviors of a mass spring damper system:

- $r = 0$  results in an undamped system which will pass the equilibrium point faster than the three other modes. The system keeps oscillating with the same frequency and amplitude for an indefinite time without converging,

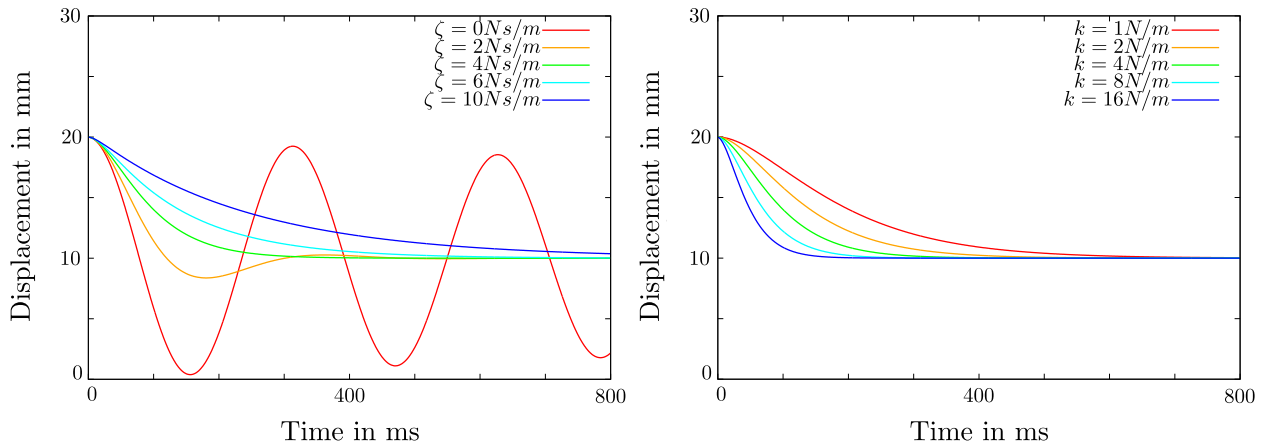


Figure 4.4.: behavior of linear mass spring damper system with an equilibrium spring length of 10 mm. The spring links a mass body of 1 kg to a fixed point in space. Initially, the spring is displaced to a length of 20 mm. Left: Variation of the damping term. Right: Variation of the spring constant for critically-damped system.

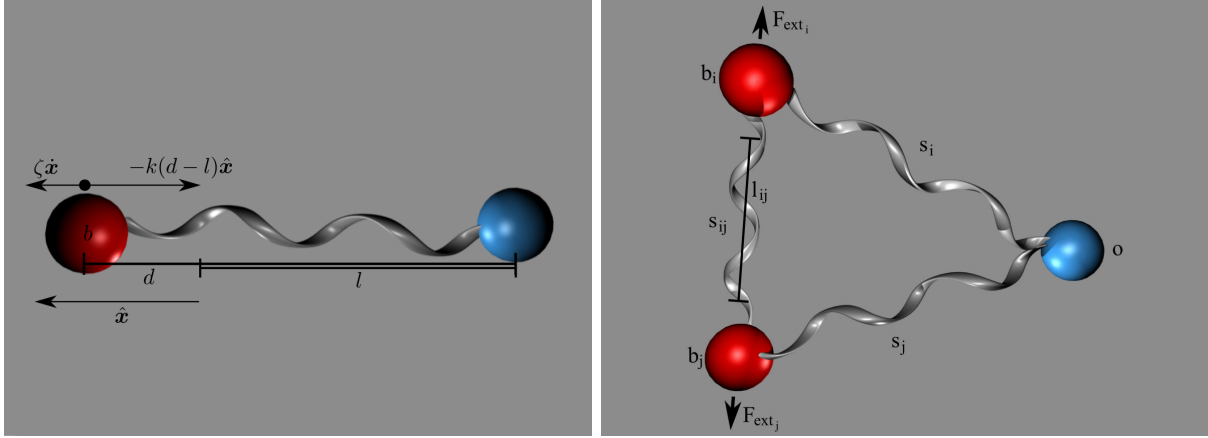


Figure 4.5.: Left: A single body linked to a fixed point in space via a spring element. Right: Multi-body systems involving two mass bodies where both are coupled via a spring and additionally attached to a fixed point in space.

- $0 < r < 1$  describes an underdamped system which will pass the equilibrium state and overshoots, but due to the introduction of frictional forces will converge to the equilibrium state while oscillating with decreasing amplitude,
- $r = 1$  specifies a critically-damped system which converges to the equilibrium state as fast as possible without oscillating,
- $r > 1$ : leads to a non-oscillating system which in addition converges slower to the equilibrium state due to high frictional forces.

As depicted in Figure 4.4, the behavior of the spring and, thus, the movement of the mass body can be varied by changing the corresponding stiffness parameter and the damping ratio whereas modifying the equilibrium spring length allows the definition of different goal states.

#### 4.2.1. Multi-Body Systems

To simulate the behavior of coherent bodies whose motion mutually influence each other, a multi-body system is defined where the interaction between the bodies is emulated using spring elements. As illustrated in Figure 4.5, for a two-body problem which involves the bodies  $b_i$  and  $b_j$  connected via a spring, the force exerted on  $b_i$  by  $b_j$  is described by:

$$\mathbf{f}_{ij} = -k_{ij}(d_{ij} - l) \left( \frac{\mathbf{x}_i - \mathbf{x}_j}{d_{ij}} \right) - \zeta_{ij}(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j) \quad (4.6)$$

with  $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$  being the distance between  $b_i$  and  $b_j$ . Analogously, the following term:

$$\mathbf{f}_{ji} = -\mathbf{f}_{ij} \quad (4.7)$$

describes the force acting on  $b_j$ . With  $b_i$  and  $b_j$  being linked to the origin of the coordinate system as defined in Eq. 4.4, the total net forces acting on  $b_i$  and  $b_j$  are described by following terms:

$$\mathbf{f}_i = \mathbf{f}_{b_i} + \mathbf{f}_{ij}, \quad (4.8)$$

$$\mathbf{f}_j = \mathbf{f}_{b_j} + \mathbf{f}_{ji}. \quad (4.9)$$

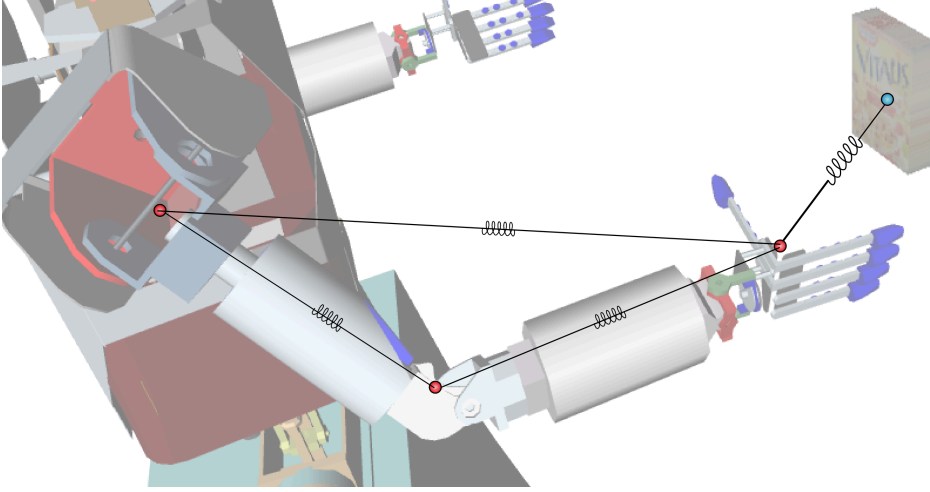


Figure 4.6.: The virtual spring concept as introduced by (Arimoto et al., 2005) illustrated for arm reaching on a humanoid robot.

Consequently, for a dynamical system with  $N$  bodies, based on the pairwise evaluation of the interactions between the bodies, the net forces acting on a single body  $b_i$  can be described using following equation:

$$\mathbf{f}_i = \mathbf{f}_{b_i} + \sum_{j=1}^N \kappa_{ij} \mathbf{f}_{ij} \begin{cases} \kappa_{ij} = 1 & , k_{ij} > 0 \\ \kappa_{ij} = 0 & , k_{ij} = 0 \end{cases}, \quad (4.10)$$

where  $\kappa_{ij}$  is a connectivity parameter which indicates whether a direct interaction between two bodies according Eq. 4.6 exists or not. With Eq. 4.4 the motion of  $b_i$  is described as:

$$\ddot{\mathbf{x}} = \mathbf{f}_{b_i} + \sum_{j=1}^N \kappa_{ij} \mathbf{f}_{ij} + \mathbf{f}_{e_i}. \quad (4.11)$$

The behavior of the multi-body system over time is determined by the equilibrium lengths and the stiffness parameters of the spring elements. The reproduction of the encoded behavior is accomplished by solving the second order differential equations which in most cases is done numerically and requires the continuous evaluation of Eq. 4.11 in equidistant time steps.

### 4.3. Virtual Spring Grasp Representation

The Virtual Spring Grasp (VSG) representation which has been proposed in (Do et al., 2011b) is based on a multi-body system. The bodies in this system represent crucial points in the task space such as the palm, the fingertips and their corresponding contact points. For the emulation of synergies as well as the encoding of prehensile movement characteristics, parameterizable virtual mass spring damper systems are incorporated. The system comprises three types of virtual springs: springs linking each fingertip with a designated contact point (see Section 4.3.1), springs between the fingertips (see Section 4.3.2), and springs anchoring the fingertips to the hand (see Section 4.3.3).

#### 4.3.1. Contact Springs

Inspired by spring-like effects in human reaching verified in numerous studies presented in (Shadmehr and Wise, 2005), several control approaches have been developed based on the

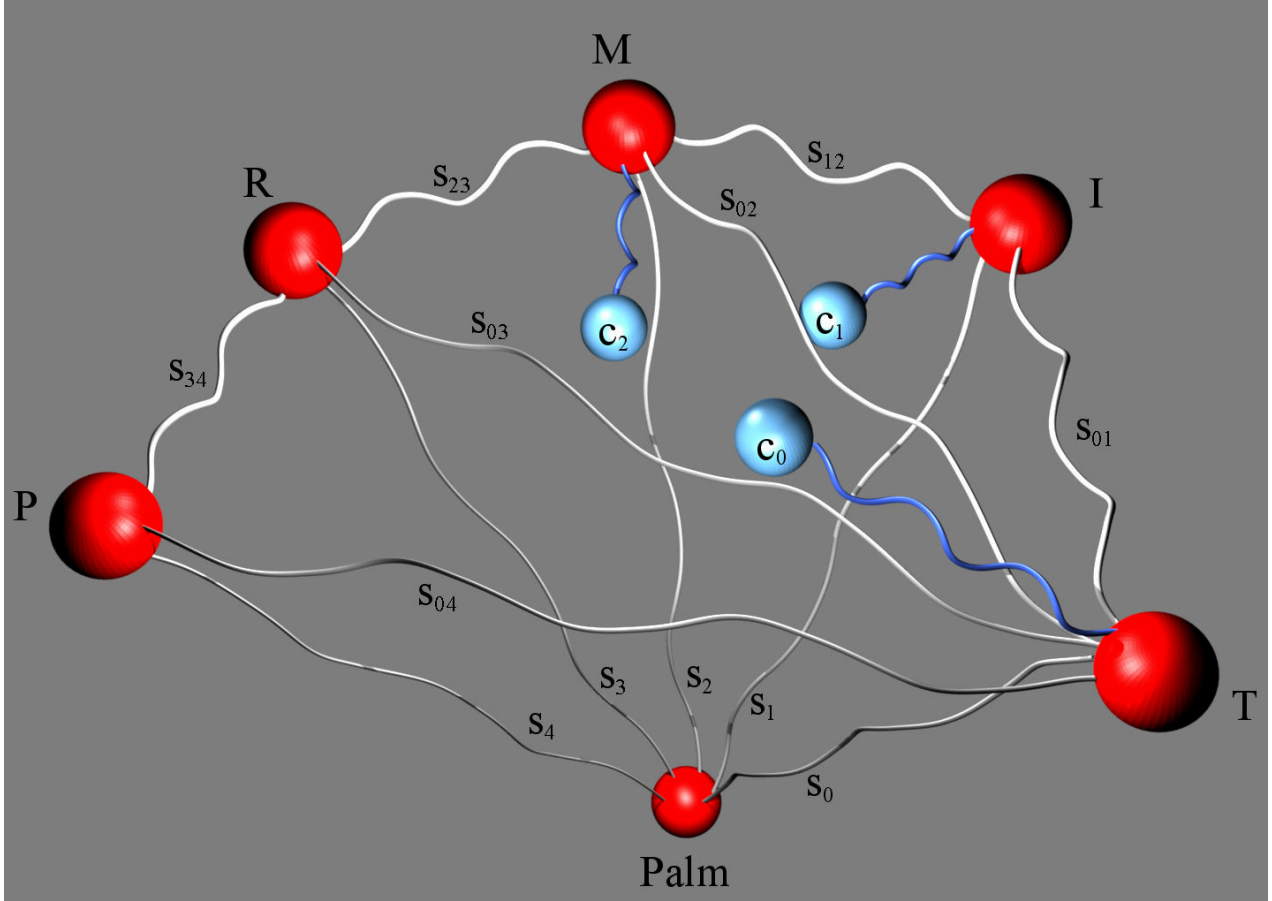


Figure 4.7.: The VSG Representation for a three-finger tripod grasp. The fingertips and the palm (red) and designated contact points (blue) are connected by the contact springs. The stabilization springs are denoted as  $s_i$  whereas the finger springs are represented by  $s_{ij}$ . The fingertips are labeled as follows: thumb (T), index finger (I), middle finger (M), ring finger (R)m and pinkie (P).

hypothesis that the relation between the endeffector and a target position can be described by a virtual spring in the task space. As depicted in Figure 4.6, (Arimoto et al., 2005) introduced a control method which in order to pull an endeffector towards its designated goal incorporates a linear mass spring system specified by a single scalar stiffness coefficient. In combination with the Jacobian matrix and a joint velocity damping term, the spring system allows the derivation of joint torques for the generation of smooth multi-joint reaching movements. A further example is given in (Bierbaum et al., 2009a) where a tactile exploration approach is presented. In order to guide a robotic hand, Virtual Model Control is applied, a control algorithm which has been originally proposed by (Pratt et al., 2001) and which uses virtual springs to simulate the robot dynamics for bipedal locomotion. In order to represent fingertip movements in a flexible and goal-directed manner, the proposed grasp representation adopts this idea by placing a virtual spring with length  $l_{vc_i}$  between each fingertip and its designated contact point. In the following, the virtual spring between the fingertip  $i$  represented by  $b_i$  and a virtual contact point  $b_{vc_i}$  is denoted as a contact spring. The position of the virtual contact point denoted as  $\mathbf{x}_{vc_i}$  emerges from the projection of the final grasp configuration  $\mathbf{x}_c \in \mathbb{R}^{3N}$  into the current hand coordinate frame  $\mathbf{x}_H(t)$ . Naturally, virtual and actual contacts points are aligned once the hand has reached the final grasp pose  $\mathbf{x}_H(T)$ . Based on the contact spring, the force  $\mathbf{f}_{vc_i}$  acting on  $b_i$  can be described as follows:

$$\mathbf{f}_{vc_i} = -k_{vc_i} (d_{vc_i} - l_{vc_i}) (\mathbf{x}_{vc_i} - \mathbf{x}_i) - \zeta_{vc_i} \dot{\mathbf{x}}_i \quad (4.12)$$

with  $d_{vc_i} = \|\mathbf{x}_{vc_i} - \mathbf{x}_i\|$  and  $l_{vc_i} \geq 0$ . In order to obtain a natural looking fingertip movement, as suggested in (Sekimoto and Arimoto, 2006),  $k_{vc_i}$  is defined as a time-varying stiffness function which is monotonously increasing and converging towards  $k_{c_i} > 0$ . Following the gamma distribution, this stiffness function is defined as follows:

$$k_{vc_i} = k_c(t) = k_{c_i}(1.0 - (1.0 + \alpha(T_{end} - t) + \frac{\alpha^2}{2}(T_{end} - t)^2)e^{-\alpha(T_{end} - t)}). \quad (4.13)$$

Similar to reaching movements, fingertip movements are assumed to be critically-damped. This assumption does not generally apply to fingertip movements, e.g. periodic movements are mostly underdamped. Nevertheless, for direct and intentional movements such as grasping, critically-damped systems provide an appropriate description of the endeffector movements aiming straight towards its goal without overshooting. Thus, considering the stiffness  $k_{vc_i}$ , the damping term  $\zeta_{vc_i}$  is determined according to Eq. 4.5:

$$\zeta_{vc_i} = \zeta_c(t) = 2 * \sqrt{k_c(t)}. \quad (4.14)$$

Since the system features a critically-damped oscillation behavior, the number of unknown variables is reduced to the spring constant parameter  $k_{c_i}$ . Hence, combined with the force term in Eq. 4.12, the movement of fingertip  $i$  is described by following differential equation:

$$\ddot{\mathbf{x}}_{c_i} = -k_c(t)(\mathbf{x}_{vc_i} - \mathbf{x}_i) - \zeta_c(t)\dot{\mathbf{x}}_i. \quad (4.15)$$

Eq. 4.15 is valid for fingertips whose contact positions have been specified. For fingertips which are not involved in a grasp e.g. the pinkie in a tripod grasp,  $k_{c_i}$  is set to zero from which  $\ddot{\mathbf{x}}_{c_i} = \mathbf{0}$  follows. The synergy study conducted in Section 4.1.1 suggests that a single parameter  $k_{c_{vf}} \in \mathbb{R}$  is sufficient for the approximation of the movements of the virtual finger  $b_{vf}$  towards the target. Since  $b_{vf}$  subsumes the fingers with the indices  $i = 2, \dots, N$ , it is assumed that  $k_{c_2} = \dots = k_{c_N} = k_{c_{vf}}$ . To accommodate the mostly decoupled thumb motion, an individual contact spring with the stiffness  $k_{c_1}$  is defined.

### 4.3.2. Finger Springs

For modeling the synergies between the fingers, a critically-damped mass spring damper system is parameterized by a spring constant  $k_{ij} > 0$  and a spring length  $l_{ij} > 0$  in order to link  $b_i$  and  $b_j$ . As suggested by the findings in Section 4.1.1, the movements between the fingers forming the virtual fingers are strongly correlated as well as the movements between the thumb and the virtual finger. Therefore, spring elements are introduced between the thumb and the fingertips forming the virtual finger in order to enforce an oppositional configuration. Furthermore, springs are installed between the each fingertip and the neighboring ones to accommodate the movement of the virtual finger as well as to allow perturbation forces acting on a fingertip to have an effect on the fingers nearby. In the following, virtual springs between the fingertips are denoted as finger springs. For a five-fingered hand ( $N = 5$ ) where the fingers are denoted as thumb ( $i = 0$ ), index finger ( $i = 1$ ), middle finger ( $i = 2$ ), ring finger ( $i = 3$ ), and pinkie ( $i = 4$ ), a spring constant matrix  $\mathbf{K} \in \mathbb{R}^{N \times N}$  is defined as follows:

$$\mathbf{K} = \begin{pmatrix} 0 & k_{21} & k_{31} & k_{41} & k_{51} \\ k_{12} & 0 & k_{32} & 0 & 0 \\ k_{13} & k_{23} & 0 & k_{43} & 0 \\ k_{14} & 0 & k_{34} & 0 & k_{54} \\ k_{15} & 0 & 0 & k_{45} & 0 \end{pmatrix}. \quad (4.16)$$

To complete the description of the entire system, a damping matrix  $\Psi$  with  $\Psi(i, j) = \zeta_{ij}$  is defined, which incorporates the damping factors for each finger spring  $s_{ij}$ . With  $\mathbf{K}$  and  $\Psi$ , the forces exerted by the finger springs and acting on  $b_i$  can be written as follows:

$$\mathbf{f}_{f_i} = -\mathbf{K}(i) \begin{pmatrix} \left(\frac{\mathbf{x}_i - \mathbf{x}_1}{d_{i1}}\right) (d_{i1} - l_{i1}) \\ \vdots \\ \left(\frac{\mathbf{x}_i - \mathbf{x}_N}{d_{iN}}\right) (d_{iN} - l_{iN}) \end{pmatrix} - \Psi(i) \begin{pmatrix} \dot{\mathbf{x}}_i - \dot{\mathbf{x}}_1 \\ \vdots \\ \dot{\mathbf{x}}_i - \dot{\mathbf{x}}_N \end{pmatrix}. \quad (4.17)$$

$l_{ij}$  is assumed to be the desired distance between both fingertips at the end of a grasping movement. Hence,  $l_{ij}$  is calculated from the last fingertip configuration of the observed grasp example as follows:

$$l_{ij} = \|\mathbf{x}_i(T) - \mathbf{x}_j(T)\|. \quad (4.18)$$

Since Eq. 4.17 is composed by the forces defined in Eq. 4.6, the movement of  $b_i$  caused by the finger springs can be encoded as following equation:

$$\ddot{\mathbf{x}}_{f_i} = \sum_{j=1}^N \mathbf{f}_{ij}. \quad (4.19)$$

### 4.3.3. Stabilization Springs

Using the spring types introduced in Section 4.3.1 and Section 4.3.2, one is able to construct a dynamical system to represent synergistic movements of fingertips approaching their contact positions. However, this preliminary system is very sensitive to perturbations causing undesirable effects such as rotation of the entire system or overshooting, which may lead to kinematically infeasible solutions. To diminish these effects, additional spring elements, which are denoted as stabilization springs, are introduced into the system. The stabilization springs anchor the bodies  $\{b_1, \dots, b_N\}$  to a single fixed point in space  $b_s$ . By defining  $b_s$  to be the origin of the local hand coordinate system and, therefore, representing the hand's palm, the movements of the fingertips are constrained according to the grasping task as well as the kinematic specifications of the hand. Based on the desired distance  $l_{s_i}$  of the finger  $i$  to the palm, a spring connecting  $b_s$  and  $b_i$  can be formally described as follows:

$$\mathbf{f}_{s_i} = -k_{s_i} (d_{s_i} - l_{s_i}) \hat{\mathbf{x}}_i - \zeta_{s_i} \dot{\mathbf{x}}_i, \quad (4.20)$$

where  $d_{s_i} = \|\mathbf{x}_i\|$  and  $\zeta_{s_i}$  corresponds to the damping factor which is calculated using Eq. 4.5. To enhance the feasibility of the solution, the kinematically reachable space is constrained by spring elements with equilibrium lengths  $l_{s_i}^{\min}$  and  $l_{s_i}^{\max}$ . The forces exerted by these springs are incorporated in the external force term  $\mathbf{f}_{e_{s_i}}$  in order to make sure that  $l_{s_i}^{\min} < d_{p_i} < l_{s_i}^{\max}$ . Subsequently, the  $\mathbf{f}_{e_{s_i}}$  is specified as follows:

$$\mathbf{f}_{e_{s_i}} = \begin{cases} -k_s (d_{s_i} - l_{s_i}^{\max}) \hat{\mathbf{x}}_i - \zeta_{s_i} \dot{\mathbf{x}}_i, & \text{if } d_{s_i} > l_{s_i}^{\max} \\ -k_s (d_{s_i} - l_{s_i}^{\min}) \hat{\mathbf{x}}_i - \zeta_{s_i} \dot{\mathbf{x}}_i, & \text{if } d_{s_i} < l_{s_i}^{\min} \\ 0, & \text{else} \end{cases}, \quad (4.21)$$

where  $k_s$  is assumed to be a constant stiffness parameter and the spring lengths  $l_{s_i}^{\min}$  and  $l_{s_i}^{\max}$  are adopted from embodiment-specific measurements of the grasping hand. Similar to the contact spring constant, the unknowns in this systems are reduced to  $k_{s_1}$  and  $k_{s_{v_f}}$  with  $k_{s_2} = \dots = k_{s_N} = k_{s_{v_f}}$  to account for the different roles of the virtual finger and the thumb in



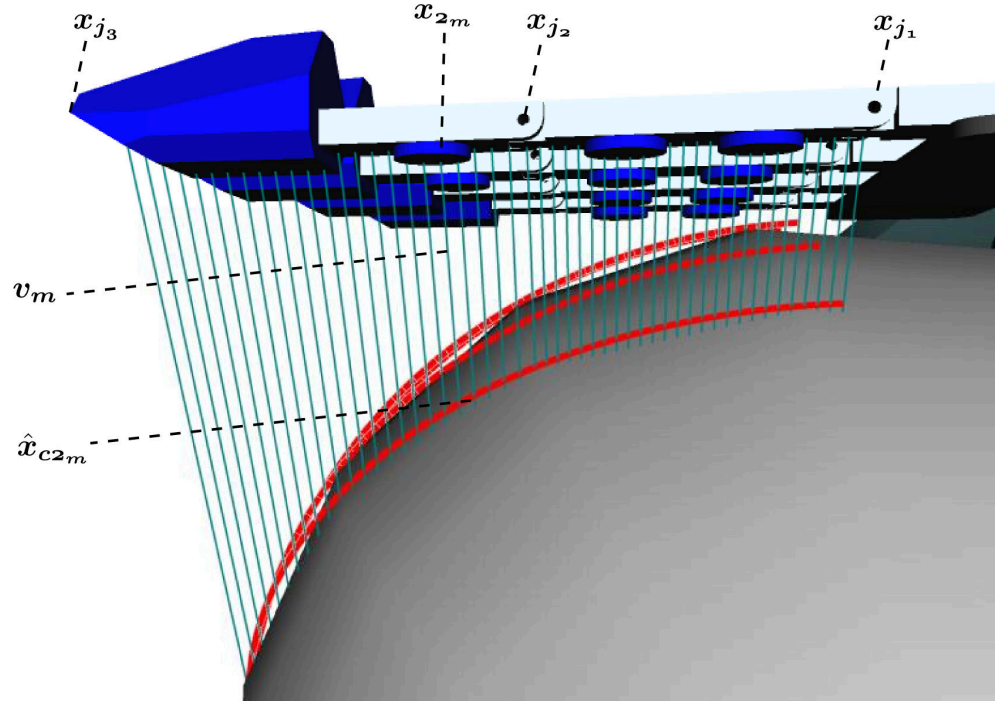


Figure 4.8.: Visualization of the Virtual Contact Strip. The finger joint positions are represented by  $\mathbf{x}_{j_i}$ . The Virtual Contact Strip is defined by points which result from the intersection of a ray originating from a point  $\mathbf{x}_{j_m}$  on the finger and the object surface.

grasping. The differential equation describing the motion of the mass body  $b_i$  concerning the forces exerted by  $b_s$  are formulated by:

$$\ddot{\mathbf{x}}_{s_i} = -k_{s_i}(d_{s_i} - l_{s_i})\mathbf{x}_i - \zeta_{s_i}\dot{\mathbf{x}}_i + \mathbf{f}_{e_{s_i}}. \quad (4.22)$$

The different spring types provide the means to encode a grasping action in a goal-directed manner. Especially, the representation of the finger-contact relationships using the contact springs facilitates the adaptation to different task- and object-specific properties. Additionally, the synergy and stabilization springs are incorporated in order to represent grasp-specific movement characteristics. The resulting dynamical system which forms the core of the VSG representation is illustrated in Figure 4.7. However, the dynamical system considers only the movements of the fingertips, and, thus, is limited to the representation of precision grasps or non-volar grasps. In the following, an extension is presented which enables the encoding of volar grasps as well.

#### 4.3.4. Virtual Contact Strip

According to the taxonomy introduced in (Kang and Ikeuchi, 1997), volar grasps are characterized by less precision but higher forces and, thus, require palm contact and, in most cases, the enveloping of the fingers around the object. The representation of volar grasps is much more complex than the representation of mere precision grasps. In order to encode and to reproduce such a grasp behavior, sufficient information about the grasping hand and the target object is needed. To represent volar grasp using the VSG representation, a concept is introduced which is denoted as the Virtual Contact Strip in order to replace the virtual contact spring described in Section 4.3.1.

Based on 3D models of the hand and the object, the Virtual Contact Strip is obtained by projecting various points belonging to finger geometry onto the object. A finger  $i$  is

described as a chain of segments  $(s_0, \dots, s_{B-1})$  and joints  $(j_0, \dots, j_{B-1})$  where  $j_0$  denotes the root joint linking the finger to the palm, the configuration of  $i$  at time  $t$  can be described by the Cartesian positions of the joints  $(x_{j_0}, \dots, x_{j_B})$  where  $x_{j_B}$  represents the position of the fingertip. Through sampling with an equidistant step size  $\nabla M$ ,  $s_b$  can be described by point sequence  $\mathbf{X}_{b_i}(t) \in \mathbb{R}^{3 \times M}$ . To determine the virtual contact point corresponding  $\mathbf{x}_{b_m}(t) \in \mathbf{X}_{b_i}(t)$ , a ray  $\mathbf{v}_m$  is constructed as follows:

$$\mathbf{v}_m = \mathbf{x}_{b_m} + \mathbf{z}_{j_0} \times (\mathbf{x}_{b+1} - \mathbf{x}_b) \quad (4.23)$$

with  $\mathbf{z}_{j_0}$  denoting the rotation axis of the 1 DoF joint  $j_0$  (in anatomical terms PIP joint).  $\mathbf{v}_m$  penetrates the object surface at the point  $\hat{\mathbf{x}}_{cb_m}$  which is considered to be the corresponding virtual contact point. Therefore, the Virtual Contact Strip can be defined as a set  $\hat{\mathbf{X}}_{cb}(t) \in \mathbb{R}^{3 \times M}$ . In order to pull the finger towards the Contact Strip, mass spring damper systems as defined in Section 4.3.1 are placed between  $\mathbf{x}_{b_m}$  and  $\hat{\mathbf{x}}_{cb_m}$  with  $l = 0$ . In the case that the ray does not intersect with the object surface no forces are applied on the corresponding finger point. Based on  $M$  force components which originate from  $\hat{\mathbf{x}}_{cb_m}$  and act on  $\mathbf{x}_{b_m}$ , the forces exerted on the fingertip are described by following term:

$$\mathbf{f}_{c_i} = \sum_{m=1}^M \ddot{\mathbf{x}}_{b_m}. \quad (4.24)$$

As mentioned before, the Virtual Contact Strip makes use of the grasping hand's geometry in order to implement an enveloping behavior and ,thus, ensures that the contact areas between object and hand are maximized. This allows the application of higher forces and increases the stability of a grasp. For now, this can only be accomplished by adapting the proposed grasp representation to the embodiment used for grasping. Therefore, a volar grasp representation cannot be transferred easily between different kinematics. However, on the other hand, the control of prehensile finger movements is facilitated since the Virtual Contact Strip allows the direct inference joint angle values.

#### 4.3.5. Enclose and Preshape

The VSG Representation is used to represent both stages of a grasping movement, the enclosing and preshaping of the hand. As depicted in Figure 4.9, two equilibrium states are defined for each stage where the states are created by adapting the spring lengths of the representation to the underlying grasp example. The ends of the enclosing and preshaping stages are denoted by  $T_{pre}$  and  $T_{end}$ . The equilibrium lengths of the finger springs and the stabilization springs are adopted from the final fingertip configuration at  $T_{end}$  and are assumed to be fixed for preshape as well as enclose. The transition between these two grasp stages is mainly accomplished by adapting the contact spring lengths accordingly. For an appropriate encoding of the preshaping movement,  $l_{vc_i} = r_{max} - r$  is specified, with  $r$  describing the object radius and  $r_{max}$  the maximal aperture which can be reached by the grasping hand. A preshaping behavior is induced into the representation which evokes repelling forces emanating from the object and acting on the fingertips. In the enclose stage,  $l_{vc_i} = 0$  is assumed which guarantees the alignment of the fingertips with the designated contact positions. Considering the two equilibrium states, the time-varying stiffness functional defined in Eq. 4.13 has to be redefined and, thus, takes the following form:

$$k_{vc_i}(t) = \begin{cases} k_{c_i}(1.0 - (1.0 + \alpha(T_{pre} - t) + \frac{\alpha^2}{2}(T_{pre} - t)^2)e^{-\alpha(T_{pre} - t)}) & , T_0 < t < T_{pre} \\ k_{c_i}(1.0 - (1.0 + \alpha(T_{end} - t) + \frac{\alpha^2}{2}(T_{end} - t)^2)e^{-\alpha(T_{end} - t)}) & , T_{pre} \leq t \leq T_{end}. \end{cases} \quad (4.25)$$

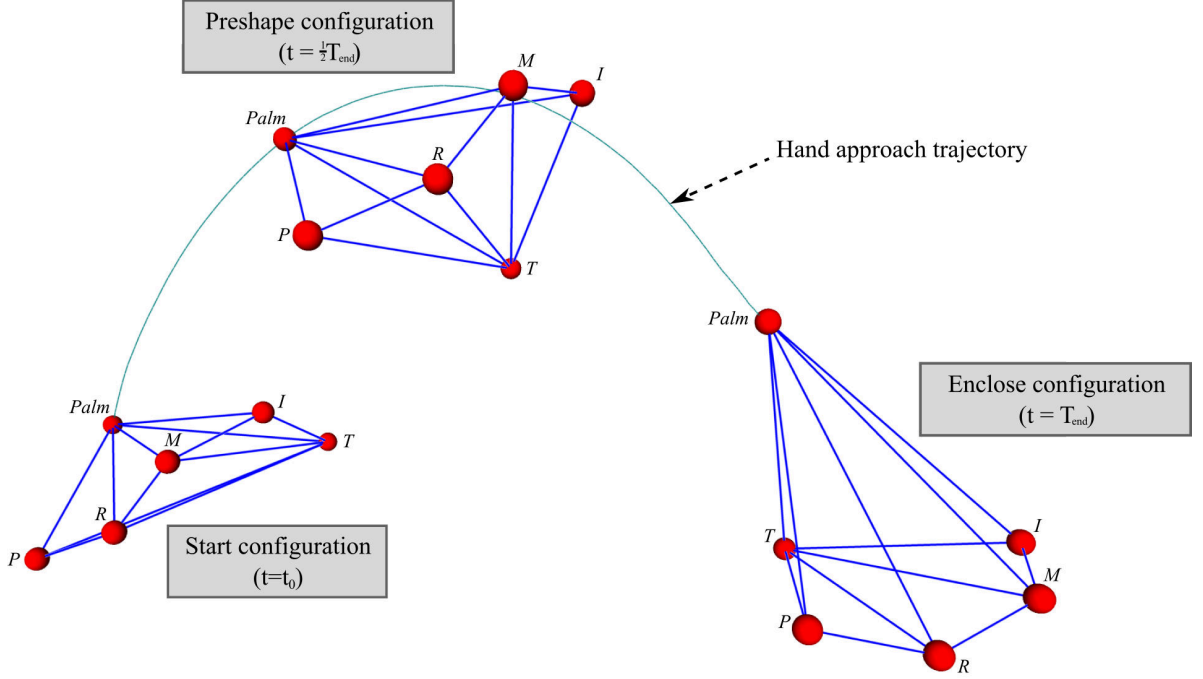


Figure 4.9.: Start, preshape, and enclose configuration of a spherical precision grasping action which is represented using the proposed approach.

The end of each grasp stage is reached once  $k_{v_i}(t)$  converged to the saturated value  $k_{c_i}$ . By introducing both functionals into Eq. 4.12, a preshaping as well as an enclosing behavior can be represented with a smooth transition between both stages. Based on the differential equations in Eq. 4.19, Eq. 4.15, and Eq. 4.22, the total motion equation for fingertip  $i$  is described as follows:

$$\ddot{\mathbf{x}}_i = \ddot{\mathbf{x}}_{f_i} + \ddot{\mathbf{x}}_{c_i} + \ddot{\mathbf{x}}_{s_i} + \mathbf{f}_e. \quad (4.26)$$

The parameters with which a specific behavior is induced consist of the spring constant matrix  $K$  and the stiffness parameters  $k_{c_1}, k_{c_{vf}}, k_{s_1}, k_{s_{vf}}$ . Since the proposed VSG representation incorporates  $M = 7$  finger springs, a representation with  $\mathbf{k} = (k_{12}, k_{13}, k_{14}, k_{15}, k_{23}, k_{34}, k_{45}, k_{c_1}, k_{c_{vf}}, k_{s_1}, k_{s_{vf}})^T \in \mathbb{R}^{11}$  is proposed for the continuous representation of a prehensile finger movements performed by a five-fingered hand.

#### 4.4. Representation of Transport

To be able to fully describe a grasping movement, the proposed representation is coupled to a model which allows the encoding of the hand approach movement. The hand approach movement can be described by models for the description of human arm reaching. It is commonly assumed that human movements are generated based on the composition of movement primitives which represent basic actions and, thus, form a vocabulary for the generation of complex movements. This assumption is supported by neuroscientific studies reported in (Wolpert and Kawato, 1998) showing that the movement control and learning behavior in humans can be explained by architectures which incorporate multiple forward and inverse models in order to compare, predict, and generate motions. In the effort to transfer these findings to robotic platforms, several approaches have been proposed to enable the extraction of such models based on movement examples. In (Ude et al., 2004) and (Wada and Kawato, 1995) demonstrated joint angle trajectories are encoded using spline-based representations. In (Tso and Liu, 1996), (Inamura et al., 2003), and (Asfour et al., 2008), approaches using

statistically-based learning methods such as Hidden Markov Models have been proposed in order to encode trajectories in the form of state sequences. The represented movements are synthesized by interpolating between these states. An alternative approach is introduced in (Calinon et al., 2007) where Gaussian Mixture Models are employed in order to generalize multiple demonstrated trajectories of the same task.

Recent research efforts dedicated to the development of endeffector movement representations focused on the use of dynamical systems. In this context, (Khansari-Zadeh and Billard, 2011) presented a method based on a motion model in the form of a multi-dimensional dynamical system which incorporates a nonlinear function composed by Gaussian basis functions. The function parameters are estimated from the demonstrations in order to enforce the system to encode the featured behavior. A popular movement representation is proposed by the concept of Dynamic Movement Primitives in (Ijspeert et al., 2002), which incorporates a linear dynamical system perturbed by a nonlinear force functional in order to encode arbitrary complex movements. A canonical system which drives the evolution of this system allows the integration and synchronization of multiple systems and, thus, facilitates the encoding of movements even in higher dimensions. In this thesis, DMPs are applied for the representation of the hand approach movement. Besides the mentioned advantages, the canonical systems enables the integration of the VSG representation into a unifying grasp representation. Hence, in the following the concept of the DMPs is described in more detail.

#### 4.4.1. Dynamic Movement Primitives

The idea behind Dynamic Movement Primitives is that each dimension of a multi-dimensional movement can be described by a simple dynamical system resembling a linear spring system which is perturbed by a nonlinear force functional. Based on this functional which is learned from demonstration a nonlinear system is derived which describes an attractor landscape with which arbitrary complex trajectories can be encoded. These systems are denoted as transformation systems. For discrete movements, a transformation system forms a unique point attractive system which is described as follows:

$$\tau \dot{\mathbf{v}} = k_H(\mathbf{x}_g - \mathbf{x}) - \zeta_H \mathbf{v} + f_H(s) \quad (4.27)$$

$$\tau \dot{\mathbf{x}} = \mathbf{v}, \quad (4.28)$$

where  $\mathbf{x}$  and  $\mathbf{v}$  are position and velocity of the system.  $\mathbf{x}_g$  is the predefined goal position and  $\tau$  denotes a temporal scaling factor with which the duration of the encoded trajectory can be controlled. The mass spring damper system described in Eq. 4.28 is parameterized with a stiffness parameter  $k_H$  and a damping factor  $\zeta_H$  which is chosen such that the system is critically-damped. For the encoding of the hand approach movement  $\mathbf{X}_H \in \mathbb{R}^D$ , the transformation systems are composed of  $D$  critically-damped spring systems. In order to learn the characteristics of a movement, for each dimension the attractor landscape is shaped based on the demonstration represented by  $(\mathbf{X}_H, \dot{\mathbf{X}}_H, \ddot{\mathbf{X}}_H)$ . To do so, a nonlinear perturbation function  $f_H$  of locally weighted models in the form of Gaussian basis functions is used:

$$f_H(s) = \frac{\sum_i w_i \psi_i(s)}{\sum_i \psi_i(s)}. \quad (4.29)$$

For each Gaussian basis function  $\psi_i(s) = \exp\left(-\frac{(s-c_i)^2}{2\sigma_i}\right)$  centered at  $c_i$  and with variance  $\sigma_i$ , an appropriate weighting  $w_i$  has to be determined using linear regression techniques. The function  $f_H$  depends on a phase variable  $s$ , which monotonically changes from 1 to 0 during a movement and is obtained by integrating following equation:

$$\tau \dot{s} = -\alpha s, \quad (4.30)$$

where  $\alpha$  is a pre-defined constant. Eq. 4.30 is referred to as canonical system. The DMP formulation features several properties such as guaranteed convergence towards the goal, spatial as well as temporal invariance and robustness against perturbations. However, the most important property lies in the simple adaptation towards new conditions such as new start and goal positions. Once specified, the execution of the movement is attained through integration and evaluation of  $s(t)$ . The obtained phase variable then drives the nonlinear function  $f_H$  which in turn perturbs the linear spring damper system to compute the desired attractor landscape. For each specific grasp type, a DMP is generated and stored along with the VSG representation in a motion library for later use.

#### 4.5. Coupling between Transport and Grip

For the synchronization of the hand approach and the grip aperture movement, the phase variable  $s$ , which evolves according to Eq. 4.30, is used to drive the temporal evolution of the VSG representation as well. Consequently, the time-varying stiffness functional described Eq. 4.13, which is the only time-variant functional within the VSG representation, is redefined with regard to the phase variable  $s$ . Therefore, in Eq. 4.13,  $t$  is replaced with:

$$t = \frac{sT_e}{\tau}. \quad (4.31)$$

With Eq. 4.31, the motion equation as defined in Eq. 4.26 is integrated in the DMP formulation as an additional transformation system.

A further option to influence the convergence behavior of the dynamical system of the VSG is provided by the external force component  $\mathbf{f}_e$ . This proves to be particularly advantageous considering eventual corrective movements which have to be performed to accommodate errors and uncertainties in the perception and execution. Specifically, the external force term defined with regard to the contact springs proves to be useful. Adding forces directed towards the contact positions accelerates the prehensile fingertip movements whereas forces directed towards the palm slow down and delay the enclosing the fingers. To attain a grasp equilibrium, it is important that for the definition of  $\mathbf{f}_e$  the condition  $\lim_{t \rightarrow T_{end}} \mathbf{f}_e(t) = \mathbf{0}$  holds.

#### 4.6. Summary

In this chapter, a continuous grasp representation defined in the task space has been presented. The definition in the task space facilitates the incorporation finger-contact relations implemented in the form of mass spring damper systems and, thus, allowing to encode goal-directed prehensile finger movements. A crucial component in the proposed representation is the emulation of finger movement synergies using virtual spring elements. The consideration of finger movement synergies allows the representation of grasp-specific characteristics with few parameters and, thus, contributes to a compact representation of coordinated prehensile finger movements. In combination with a representation for the hand approach, a grasp primitive is formed which can be adapted to the different task- and object-specific properties. Both representations are coupled via dynamical systems. Regarding the learning of grasp primitives, the proposed representation which is based on a strong simplification of the hand's structure is used to facilitate the observation, learning, and the reproduction of grasping actions. However, it is stressed that the incorporated model lacks essential details about

#### 4. Grasp Representation

---

the kinematics and dynamics of the grasping hand and does not consider dynamic object properties such as weight and friction. Therefore, the proposed approach is used to represent the grasping kinematics emerging from grasp examples observed in the task space.

## 5. Grasp Data Acquisition

In the previous chapter, a model has been proposed which allows the encoding of anthropomorphic grasps and, thus, enables the robot to learn grasping skills from human grasp examples. To accelerate and facilitate the grasp learning process, grasp examples are obtained through the observation of grasp demonstrations performed by the human. Human grasp demonstrations feature highly optimized movements which are well adapted to the environment and the objects to be grasped. Thus, the observation of humans is a way to acquire ideal grasp data for the generation of situation-specific grasping strategies. In this chapter, methods will be discussed which allow the capturing of human grasping movements using optical sensors. For the creation of grasp data, markerless and marker-based methods have been employed. In Section 5.1, advances in this field of human motion capture is reviewed. In Section 5.2, the marker-based motion capture system used in this work is described whereas the developed markerless approach is presented in Section 5.3.

### 5.1. Human Motion Capture with Optical Systems

Capturing of human motion has been extensively studied in the last decades leading to numerous methods and systems which are applied in various fields such as robotics, sports, medical, and entertainment applications, just to name a few. Especially, optical systems are a focus of interest since with these systems human movements can be registered without the human subject being physically confined to the system. Furthermore, most optical motion capture systems allow the observation of large areas and, thus, enable the interpretation of the performed motion in situation-specific contexts. Using multiple calibrated sensors image features representing characteristic human body parts are captured. Based on triangulation the 3D positions of these body parts are calculated. In general, optical systems are categorized based on whether physical markers or natural image features are used to denote these body parts. Thus, these human observation techniques can be subdivided in two categories: marker-based and markerless.

Marker-based human motion capture systems are popular in commercial applications. The usage of physical markers allows the generation of prominent signals and features which can be rapidly processed. By focusing solely on these features, using such systems an arrangement of multiple markers can be reliably localized and tracked. Based on this property, human movements can be captured with an outstanding performance in terms of accuracy, frequency, and robustness. However, most marker-based methods require extensive and costly equipment. Thus, a motion capture procedure entails considerable effort. For example, preparatory work has to be performed which includes the accurate attachment of markers on designated positions on the human subject and objects of interest. Furthermore, slight changes in the configuration of the integrated sensor devices require a recalibration of the entire system. Hence, marker-based motion capture are mostly bound to a fixed location and used for the acquisition of motion data.

As opposed to this, markerless methods mostly require much less and simpler hardware. In many cases, sensors integrated in most robotic platform such as a stereo camera setup can be used to obtain information sufficient for the detection and the tracking of human movements. Therefore, using markerless methods enables the capturing of human motion

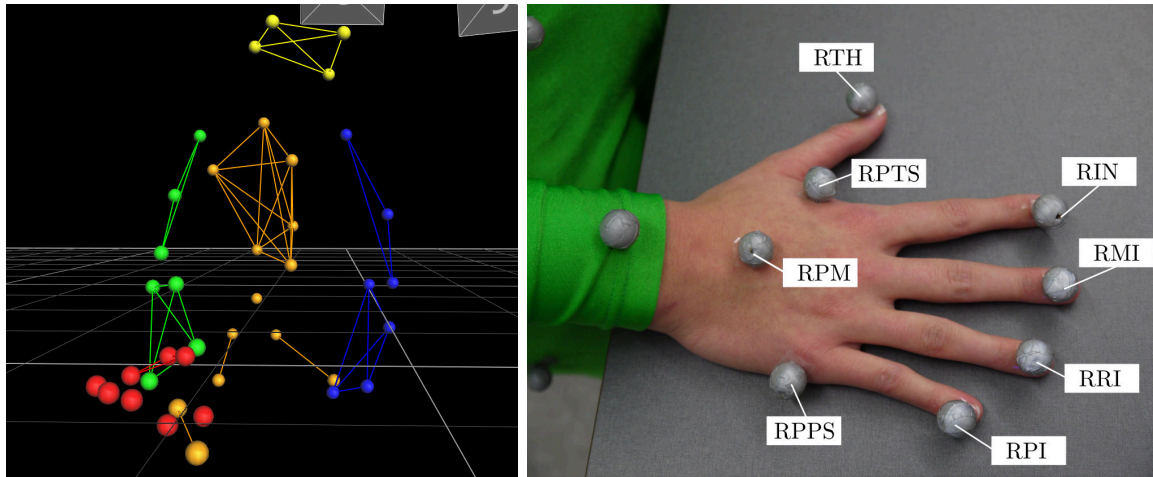


Figure 5.1.: Left: Visualization of the Vicon model used for human motion capture showing a pinch grasp demonstration on a stick. Right: The marker placement defined for the observation of grasping movements performed by the right hand.

invariant of time and location and, thus, allowing the implementation of an intuitive and natural interface which can be used to empower robots to observe and learn actions in an online manner. However, the restrictions of the used sensors lead to inaccuracies caused by erroneous computations and occlusions which, subsequently, result in noisy motion data. In addition, due to lack of markers, markerless methods rely on the extraction and processing of common image features which have to be correctly interpreted in order to put the observation into the movement context. The image processing line limits the frame rate with which a human motion can be captured.

## 5.2. Marker-based Motion Capturing

One of the most popular commercially available systems is provided by [Vicon Motion Systems]. The technique, which is used here, relies on infrared cameras and artificial reflective markers. The markers are placed on predefined body parts of a human subject. In a defined workspace, the subject is surrounded by a set of infrared cameras. Each camera is equipped with an infrared strobe, emitting a light signal, which is reflected by the markers. The reflected light, which distinguishes itself from the background, is registered by the cameras. The data from each camera consisting of 2D coordinates of each recognized marker position, is merged in a data station, which computes the 3D position by triangulation and the label of each visible marker. The integrated cameras have a resolution of 2 megapixels at 500 Hz which allows the tracking of marker movements at frame rates up to 2,000 Hz. Besides the hardware, the system contains a comprehensive software package, which facilitates the calibration and handling of the system. Due to the high-speed and high-resolution properties of the cameras, the Vicon system provides an accurate method for capturing human motion at high frame rates. Furthermore, since the use of numerous markers allows capturing of barely visible motion of unobtrusive joints, complex kinematic models are applicable for the processing and representation of the motion data. The problem of occlusion of body parts is reduced to a minimum, since multiple cameras are used, which deliver multiple views of the same subject. However, occlusions occur and can cause markers to disappear temporarily. In these cases, tracking of the disappearing markers is discontinued. Therefore, the data requires a manual post-processing in order to remedy these defects. Nevertheless, human motion capture using optical markers is the method of choice if highly accurate data is needed.



To obtain grasp examples from human observation, in this thesis, it has been focused on the capturing of the fingertips and the hand's back disregarding finger segments. Hence, to reduce the risk of being occluded and increasing the robustness of the capturing procedure, larger markers have been placed on the fingertips. In addition to the prehensile finger movements, markers have been attached on the upper body in order to gain information about the hand approach movement. The marker positions for the upper body have been determined from (Stein et al., 2007). A snapshot of the human motion capture process as well as an illustration of the marker arrangement on the grasping hand is depicted in Figure 5.1.

### 5.3. Markerless Motion Capturing

Markerless human motion capture has been a research topic which attracted a lot of attention in recent years and which, at least in daily life, climaxed in the development of sensor systems such as the Microsoft Kinect sensor dedicated to this purpose. In the field of humanoid robotics, the observation of humans becomes particularly challenging due to the onboard sensor capabilities which are limited regarding the resolution, the frame rate, and the field of view. In the context of grasping, the real-time observation of simultaneous arm and finger movements cannot be achieved with a single method, but rather with a combination of various methods for the observation of specific body parts. Hence, in this thesis, for the observation of human grasping movements, a framework is proposed which incorporates a method for the markerless tracking of the human upper body as well as a fingertip tracking approach. First, in Section 5.3.1, related approaches in the field of markerless human motion capture are reviewed. Subsequently, the methods integrated in the proposed framework are described.

#### 5.3.1. Advances in Markerless Human Motion Capture

##### Upper Body Tracking Approaches

Numerous approaches have been proposed for the tracking of coarse human full-body and upper-body movements. The methods used for tracking are tailored to the used sensor systems.

In (Demirdjian et al., 2003), a system is presented which is based on the Iterative-Closest-Point (ICP) algorithm. The ICP algorithm is used to fit rigid bodies of a 3D model to a depth map calculated from a stereo camera image pair. The model configuration which consists of the positions and orientations of the bodies is projected onto an articulated motion space in order to enforce joint and contact constraints. The resulting configuration serves as an input to a previously trained SVM model with which a previous body posture is mapped to a new valid body posture. An ICP-based approach using depth maps which are captured with an active 3D sensor is presented in (Knoop et al., 2009). For tracking, a model is used which consists of cylinders where each cylinder represent a body segment. To enforce joint constraints point correspondences between these cylinders are modeled in the form of elastic bands. Considering these constraints, the ICP algorithm is applied on each body segment separately in order to fit the model to the current point cloud. In (Bregler and Malik, 1998), a human pose estimation and tracking method is introduced which formulates the problem as minimization problem based on 2D-3D correspondences. Rigid segments which form a 3D model of the human body are projected and modeled as ellipsoids on the image plane. The kinematic chain of the model is specified by twists and exponential maps. Using an EM algorithm, a model configuration is searched for which the distance between the modeled segments and edge pixels describing the contour of the human subject becomes minimal. In (Wachter and Nagel, 1999), an approach is presented for model-based tracking in monocular image sequences. An Iterated Extended

Kalman Filter is used to estimate the model parameters of a 3D model consisting of elliptic cones by minimizing contour and region information between the model projection and the actual image.

Methods based on minimization do not scale well in high-dimensional parameter space causing the tracking to get stuck in local minima and, thus, making it difficult to obtain an optimal estimation for the model parameters. To address these deficiencies, in (Deutscher et al., 2000), an approach based on particle filtering is introduced for the capturing of human movements. Using the particle filter algorithm instead of merely optimizing a model configuration a search for the optimal model parameter is performed which allows the tracking method to escape from local minima. Edges and regions are detected by cameras placed around the human subject to be tracked. The particles consist of model configuration hypotheses which are weighted according to edge and region cues determined based on the comparison between projections of the 3D model on the corresponding image planes and detected image features. Similar to this approach, (Azad et al., 2008) presented a particle filtering tracking method which relies on stereo camera images. In addition to the edge cues, a distance cue is defined based on a stereo-based 3D hand and head tracking. To increase the robustness, the method implements a particle sampling in the vicinity of an inverse kinematics solution in order to facilitate the reinitialization of the procedure. In the proposed framework, the method presented in (Azad et al., 2008) is used for the capturing of human arm reaching movements. Therefore, this approach is subject to a more detailed discussion in Section 5.3.3.

### **Finger Tracking Approaches**

The hand is considered to be one of the most crucial body parts regarding the interaction with other humans and the environment. Especially in the field of humanoid robots where machines share similar kinematic structures such as humans e.g. a five-fingered hand, an efficient method for the tracking of finger movements enables a robot to obtain relevant information on human grasping. For this purpose, full hand tracking approaches in joint angle space have been proposed in for example (Rehg and Kanade, 1994), (Stenger et al., 2001), and (Oikonomidis et al., 2010). However, due to the highly complex structure of the hand whose motion involves at least 21 DoF, tracking of the hand can be only achieved with high costs which in return results in a low frame rate. Using mere stereo vision, a reasonable solution lies in the reduction of the problem's dimensionality by shifting from joint angle space into the task space where fingertips provide prominent features which in most cases contain sufficient information for further post-processing. In (Blake and Isard, 2000), a finger tracking approach based on Active Contours is presented for the scenario of air-writing. The target to be tracked consists of a contour which is laid around the pointing finger. The finger posture is assumed to be fixed, thus, no reliable statement can be made on the actual fingertip position. In (Argyros and Lourakis, 2006), fingertips are detected within a contour which is extracted from skin blob tracking based on curvature properties. A more elaborate approach is presented by (Hsiao et al., 2008) using Particle Random Diffusion where particles are propagated from the center of the hand to positions close to the contour. The intersection of the contour with predefined line segments centered at a particle and the examination of the resulting transitions from non-skin area to skin-area and vice versa indicate whether a particle is considered to be a fingertip. This method does not take into account possible overlapping of fingers with the palm region and is specially designed to detect tips of stretched fingers. Based on multi-scale color features, (Bretzner et al., 2002) introduces a hierarchical representation of the hand consisting of differently sized blobs where each blob represents a part of the hand. The blob features are matched with a number of hierarchical 2D models where each model incorporates

a specific finger pose. Based on the limitation that finger postures are fixed, the tracking is accomplished by applying a particle filtering technique. In order to implement a continuous fingertip tracking method, one should rely on prominent features which can be extracted at any time of an image sequence. In (Burns and Wanderley, 2006), for detecting a guitarist's fingertips, circular features are proposed which are localized by performing a circular Hough transform. A similar approach is introduced in (Kerdvibulvech and Saito, 2008) where fingers are located based on semi-circular templates.

### 5.3.2. Particle Filter Framework

The particle filter, originally introduced as the CONDENSATION algorithm in (Isard and Blake, 1998), is a Bayesian filtering technique based on sequential importance resampling with which a posterior probability density function of a state vector can be approximated from a finite set of weighted samples, respectively particles, drawn from a probability distribution. A particle can be described as  $(\mathbf{s}, w)$  with  $\mathbf{s}$  denoting a state hypothesis and  $w$  the corresponding likelihood.  $w$  is determined based on how accurate the state estimate  $\mathbf{s}_t$  matches the current observation  $\mathbf{o}_t$  according to likelihood function  $p(\mathbf{o}_t|\mathbf{s}_t)$ . The approximation of the posterior probability density function is based on a set of  $M$  particles which is denoted as  $X_t = \{(\mathbf{s}_t, w_t)\}_{i=1, \dots, M}$ . Thus, the particle estimation of the current state can be formulated as follows:

$$\hat{\mathbf{s}}_t = \sum_{i=1}^M w_t^i \mathbf{s}_t^i. \quad (5.1)$$

To propagate the approximation over time, the particle filter algorithm is embedded in an iterative scheme. In each iteration,  $m$  particles which are drawn from  $X_{t-1}$  are resampled and in order to form set of state hypotheses  $\mathbf{s}_t$ . Then, each hypothesis is evaluated according to  $p(\mathbf{o}_t|\mathbf{s}_t)$ . State hypotheses and weights are combined to particles which form  $X_t$ . As suggested in (Isard and Blake, 1998) for the sampling of the particles, a dynamic model can be taken into account which provides predictions on future locations given the current state and the previous state. In addition, to account for unpredictable movements Gaussian noise can be added. Adding Gaussian random variable  $\Psi$  and a displacement vector  $\hat{\mathbf{v}}_j$ ,  $\mathbf{s}_t^i$  can be written as:

$$\mathbf{s}_t^i = \mathbf{s}_{t-1}^j + \hat{\mathbf{v}}_t + \Psi. \quad (5.2)$$

### 5.3.3. Upper Body Tracking

For the capture of the human arm reaching movements, a real-time stereo-based human motion capture system which has been introduced in (Azad et al., 2008) is applied. Based on a stereo color image sequence, the system allows the tracking of the human upper body using an adapted 3D model. The upper body model depicted in Figure 5.2 incorporates 14 DoF (6 DoF for the base transformation, 2-3 for the shoulders, and 2-1 for the elbows). The shoulder is modeled as a ball joint and the elbow as a hinge joint. Consisting of rigid body parts representing the torso, the upper arm, and the lower arm, the model provides a simplified description of the kinematic structure of a human. These body parts are visualized using cones which are scaled to fit the segments of the human subject to be observed. The model tracking is implemented using the particle filter algorithm which uses two different cues, an edge and a distance cue, in order to evaluate how well a given model configuration matches the current observations. Regarding the edge cue, observations are extracted from stereo images in the form of an edge map  $I_e$ . A particle  $\mathbf{s}_t^i$  which denotes a configuration for the model to be tracked is transformed into a set of contour points  $C$  with  $C = \{c_1, \dots, c_P\}$  consisting of image

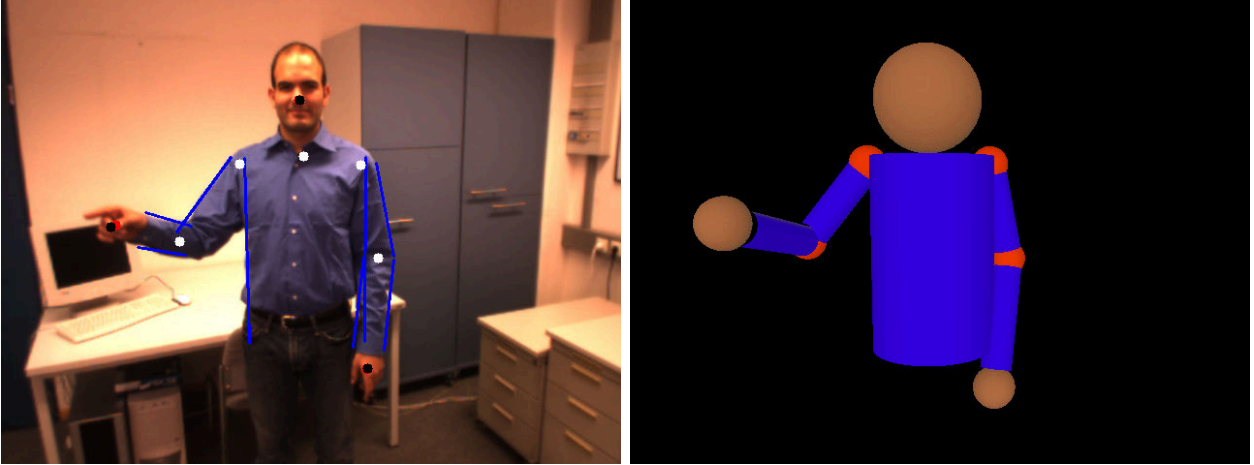


Figure 5.2.: Left: Upper body model projected on the image plane. Right: Visualization of the 3D model used for upper body tracking using the method proposed in (Azad et al., 2008). (©2008 IEEE.)

coordinates  $c_p = (x_p, y_p)$  obtained by projection of the model onto the image plane. Based on a comparison between  $I_e$  and  $C$ , the edge cue is defined as follows:

$$p_e(I_e | \mathbf{s}_t^i) \propto w_e(\mathbf{s}_t^i) = 1 - \frac{1}{P} \sum_{p=1}^P I_e(x_p, y_p). \quad (5.3)$$

Using the distance cue, each model configuration is evaluated according to the distance between points representing common body parts of the human subject and the 3D model. These points represent hand and head position of the model and the human. A skin color tracking method is used to localize candidate areas in the image  $I_d$  which are labeled as head and hand according the relative arrangement of these body parts. Corresponding model points are calculated using forward kinematics on the 3D model with the pose  $\mathbf{s}_t^i$  which are combined in  $B = (p_0, p_1, p_2)$ . Similar to the edge cue, the distance is defined as follows:

$$p_d(I_d | \mathbf{s}_t^i) \propto w_d(\mathbf{s}_t^i) = \sum_{b=1}^B \|p_i - p_i(I_d)\|. \quad (5.4)$$

The final likelihood function is constructed from Eq. 5.3 and Eq. 5.13 yielding following equation for the computation of the weights:

$$w_t^i = \frac{\sqrt{w_e(\mathbf{s}_t^i) w_d(\mathbf{s}_t^i)}}{\sum_{k=1}^M \sqrt{w_e(\mathbf{s}_t^k) w_d(\mathbf{s}_t^k)}}. \quad (5.5)$$

To obtain a current state estimate  $s_t$  for the human model configuration the sum of all particles is evaluated as stated in Eq. 5.1. A deficiency of this approach lies in the exponential growth of the search space with increasing dimensionality. For an accurate state estimation, high-dimensional problems require a sufficiently large number of particles. To increase the robustness and efficiency of the method, various extensions have been introduced such as a prioritized fusion method, adaptive shoulder positions, and the incorporation of the solutions of the redundant arm kinematics which help to constrain the search space.

### 5.3.4. Hand Tracking

The previously described tracking framework operates on peripheral views in which the entire upper body is visible. Due to the limited resolution of the cameras, the views do not

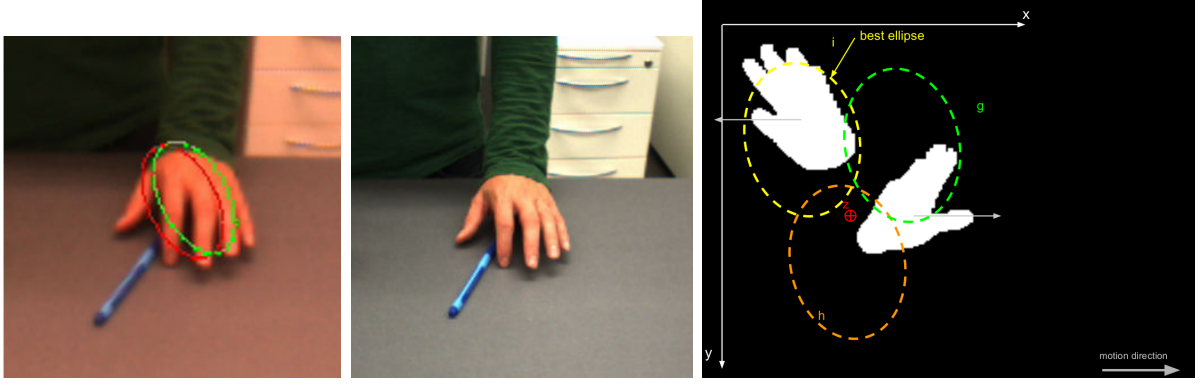


Figure 5.3.: Left: Peripheral view of the grasping hand. Center: Foveal view obtained during hand tracking. Right: Visualization of hand tracking particles in the form of ellipses.

contain sufficient data based on which motion information about the finger movements can be extracted. To obtain more detailed views on the grasping hand, foveal views are captured and used for the hand tracking procedure. For the computation of the distance cue, the upper body tracking framework incorporates a hand tracking algorithm which describes each state of the hand  $\mathbf{s}_t^i$  by its position. To increase the robustness of the hand tracking method without risking any severe performance losses, the existing method is extended by simple shape features. Hence, the hand in the image plane is described by an ellipse  $e_I = (\theta_I, r_I)$  with  $\theta_I$  denoting the orientation of the ellipse and  $r_I$  the ratio between the minor and the major axis  $a_I$  and  $b_I$ . Based on the ellipse features, in addition to the distance cue, a shape cue is defined as:

$$p_s(e_I | \mathbf{s}_t^i) \propto w_s(\mathbf{s}_t^i) = \frac{1}{2} \left( \frac{|\theta_i - \theta_I|}{\pi} + |r_i - r_I| \right). \quad (5.6)$$

By replacing  $w_e$  with  $w_e$  and redefining  $w_d$  for  $B = 1$  in Eq. 5.5 the likelihood function for the hand state estimation is derived.

### 5.3.5. Fingertip Tracking

In order to obtain data from which grasp primitives can be learned, a fingertip tracking framework is proposed allowing the observation of fingertip movements in the task space. The localization and tracking of fingertips is a difficult problem due to lack of prominent features and the frequent overlap of the finger and the palm. Based on foveal views focusing in the grasping hand, the approach is proposed in (Do et al., 2011a) where fingertip positions are determined based on circular image features. These features are tracked using a method which combines particle filtering with a mean shift algorithm. To enhance the feature extraction process an edge map is generated by applying a multi-scale edge extraction technique which is described in Section 5.3.5. Based on this map, circular image features are extracted and formed into observations which are needed for the actual tracking algorithm proposed in Section 5.3.5. To increase the robustness of the algorithm, dynamical motion models are trained for the prediction of the finger displacements in Section 5.3.5.

### Feature Extraction

In order to generate the edge image, a skin color segmentation is performed for extracting the hand and finger regions. Morphological operators are applied on the segmented image to eliminate noise and to produce a uniform region. To detect the edges in this preprocessed

image, a filtering method on various scales is used. Details on the filtering step and the subsequent Hough transformation procedure are given in the following.

**Multi-Scale Edge Extraction** Considering the problem of fingertip tracking, due to small intensity variances between different parts of the hand, e.g. the fingernail and the skin, respectively, the finger regions and the palm, it is desired to detect edges where contrast can vary over a broad range. Depending on the parameters, applying standard algorithms, such as the Canny edge detectors on a wider scale, leads to an edge image where numerous, false edges occur. To preserve low contrast edges in certain areas while reducing noise close to high-contrast edges, based on the work of (Elder and Zucker, 1996), a filter approach has been implemented consisting of a steerable Gaussian derivative filter on multiple scales. The basis filters are defined as follows:

$$\begin{aligned} G_k^x(\sigma_k; x, y) &= \frac{-x}{2\pi\sigma_k^4} e^{-\frac{(x^2+y^2)}{2\sigma_k^2}} \\ G_k^y(\sigma_k; x, y) &= \frac{-y}{2\pi\sigma_k^4} e^{-\frac{(x^2+y^2)}{2\sigma_k^2}}. \end{aligned} \quad (5.7)$$

To define the scale at which the gradient can be reliably estimated, the magnitude of the filter response  $r_k^x(x, y; \sigma_k)$  and  $r_k^y(x, y; \sigma_k)$  obtained by convolution of the image  $I$  with the filters is checked against a noise threshold. While the magnitude can be calculated according to:

$$r_k^m(\sigma_k; x, y) = \sqrt{r_k^x(x, y; \sigma_k)^2 + r_k^y(x, y; \sigma_k)^2}, \quad (5.8)$$

the threshold is set by the following function:

$$c_k = \frac{\sqrt{-2\ln(1 - (1 - \alpha)^R)}}{2\sigma_k^2\sqrt{2\pi}} s_l \quad (5.9)$$

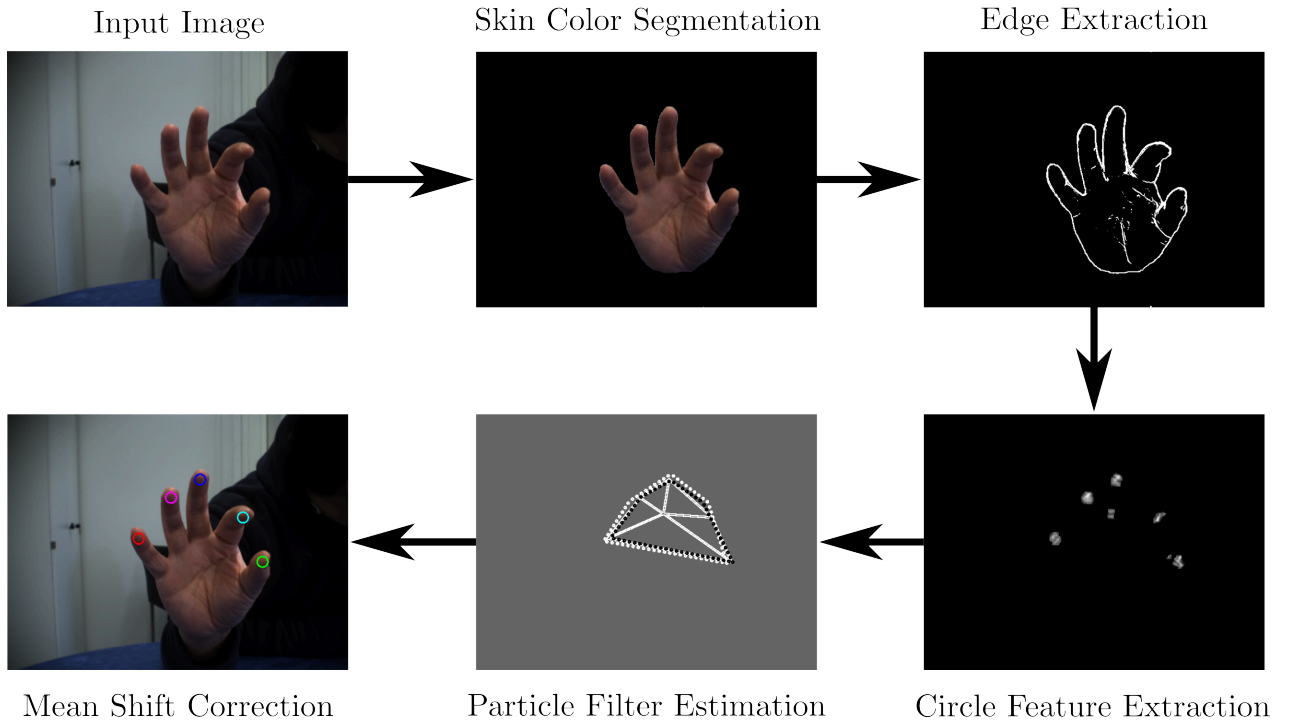


Figure 5.4.: Processing line of the proposed fingertip tracking approach.



Figure 5.5.: Left: Original input image. Center: Edge image using Canny detector. Right: Edge image using the method proposed in Section 5.3.5.

with  $s_l$  representing the standard deviation and  $\alpha$  the significance level for an image with  $R$  pixels which defines an upper boundary for allowed misclassification of image pixels. To take into account local intensity and contrast conditions, it is focused on local signal noise in a specific region rather than on global sensor noise. Therefore, Eq. 5.9 depends on the local standard deviation  $s_l$  calculated within  $2\sigma_k^{max} \times 2\sigma_k^{max}$ -neighborhood where  $\sigma_k^{max}$  denotes the largest scale being examined. Hence, each gradient at the minimum reliable scale  $\sigma_k^{min}$  is calculated where the likelihood of error due local signal noise falls below a standard tolerance. This guarantees that a more accurate gradient map is estimated which is less sensitive to signal noise and errors caused by interference from nearby structures. Based on the resulting map, the edges are extracted and with non-maximum suppression, one obtains an edge image which is depicted in Figure 5.5.

**Hough Transformation for Circle Detection** The circle features representing a certain fingertips  $n \in \{1, \dots, N\}$  are detected by applying a Hough transformation with radius  $r$ . For each edge point  $(x, y)$  with known direction in the form of a rotation angle  $\theta$ , a vote is assigned to possible circle feature positions  $(u, v)$  in two-dimensional Hough space  $I_H$  according to:

$$I_H(u, v) = I_H(u, v) + 1 \quad (5.10)$$

with

$$\begin{aligned} u &= x \pm r \cos(\theta) \\ v &= y \pm r \sin(\theta). \end{aligned} \quad (5.11)$$

Unfortunately, curves around the fingertips do not always feature perfect circular arcs. To cope with noisy and slightly deformed curves, the voting is performed for a set of radii  $R_n = \{m_0 r, \dots, m_n r\}$ . The radius  $r$  is scaled by factors  $0 < m_0 < m_n < 2$ . In order to increase the robustness of the tracking algorithm, a density distribution is formed in Hough space by convolving  $I_H$  with a Gaussian kernel  $G(\frac{r}{2}; u, v)$ .

Since the hand motion occurs in the task space, for proper fingertip tracking, a fixation of  $r$  is only valid if movement of the fingertip in direction of the z-axis of a camera is excluded.

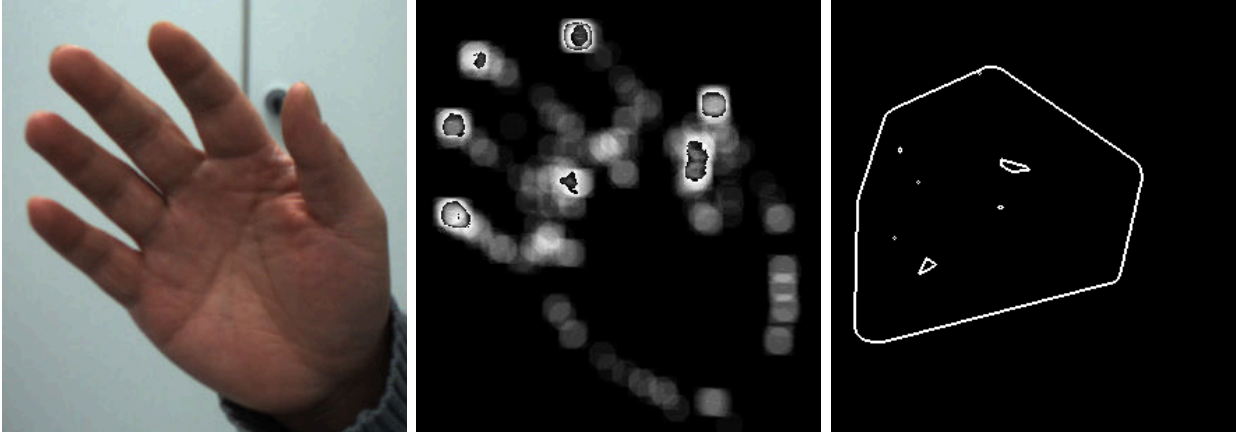


Figure 5.6.: Left: Original input image. Center: Visualization of the Hough space. Right: Generated contour for the particle filter tracking.

Adaptation of  $r$  in each frame, allows fingers to be tracked in all directions. Based on the generated density distribution, for fingertip  $n$  in frame  $t$ , a radius estimate  $\hat{r}_t$  is determined by applying an Expectation Maximization algorithm. Further details are given in Section 5.3.5.

### Initialization

The tracking procedure is initialized based on the assumptions that the hand is opened, the palm is visible, and the hand is not rotated. Based on these restrictions, the hand's palm is localized and possible finger regions are localized using the Hough transformation with different radii constructed with  $m = 3$ . The maximum bins  $I_H^i$  with  $i = 1 \dots N$  are labeled according to the assumption that, for the right hand, the thumb is represented by the right-most bin in the image. Using a line sweep method in polar space the remaining fingers are labeled accordingly (for the right hand the line sweep is performed counter-clockwise leading to the fingertip labeling  $n = \{Thumb = 0, Index = 1, Middle = 2, Ring = 3, Pinkie = 4\}$ ). Analogously, for the left hand, the thumb is located at the left-most bin and the line-sweep is performed clockwise.

### Particle Filter Tracking

For the proposed fingertip tracking framework, a state  $s$  consists of the  $N$  fingertip positions of the hand with each position being denoted by the coordinates  $(x, y)$  within the image. The approximation of  $p(z_t | s_t)$  is based on two cues: a contour and a distance cue. The contour cue is derived by exploiting the external energy functional  $E_{img}$  of a contour  $C_t^i$ . The contour is obtained by connecting the single points in  $s_t^i$  according to the finger order. The  $E_{img}$  is determined in terms of an edge image  $Z_t^E$  which is constructed by drawing lines between a set of maximum bins  $Z_t^V$  that can be found in  $I_H$ . As a result, the likelihood function can be written as:

$$p_c(z_t | s_t) \propto w_c(s_t) = \exp\left(\frac{-E_{img}(Z_t, C_t)}{\sigma_c^2}\right). \quad (5.12)$$

The distance cue is calculated from the Euclidean distance between  $s_t^i$  and  $Z_t^V$  which consists of the sum of minimal distances between  $s_t^i(j)$  and  $Z_t^V$ . Based on this cue, the likelihood function can be defined as

$$p_d(z_t | s_t) \propto w_d(s_t) = \exp\left(\frac{-\sum_{j=1}^N \min(\|s_t(j) - Z_t^V\|)}{N\sigma_d^2}\right). \quad (5.13)$$



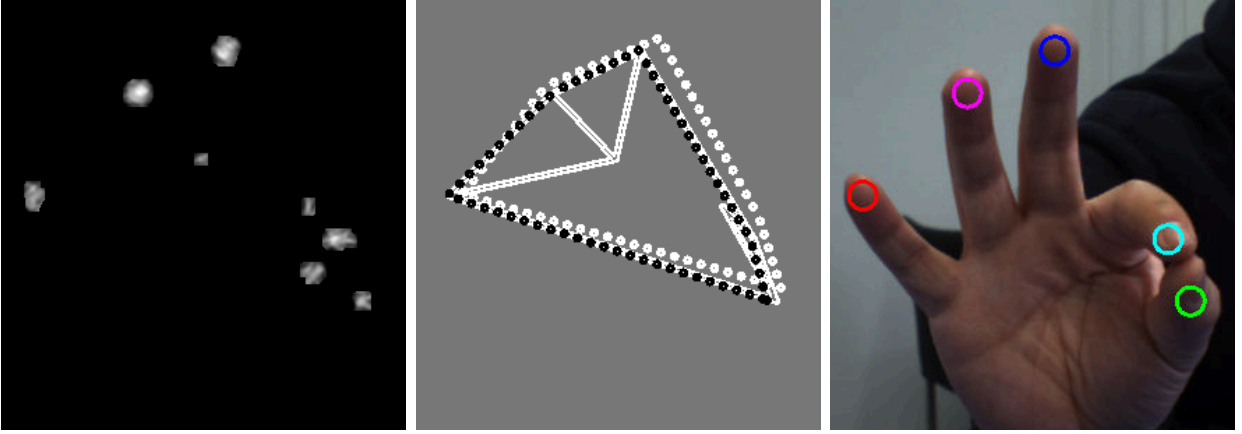


Figure 5.7.: Left: Filtered Hough space. Center: Particles in the form of contours connecting the fingertip candidates. The black contour represents the best particle whereas the white contour is weighted lowest. Right: Result image.

The final likelihood function is constructed from Eq. 5.12 and Eq. 5.13, hence, the computation of the weights is defined as follows:

$$w_t^i = \frac{\sqrt{w_c(s_t^i)w_d(s_t^i)}}{\sum_{k=1}^M \sqrt{w_c(s_t^k)w_d(s_t^k)}}. \quad (5.14)$$

To obtain a current state estimate of the fingertip configuration the sum of all weighted particles is evaluated.

### Mean Shift Correction

To obtain more accurate position estimations, a mean shift algorithm is applied to move the estimated fingertip position  $p_n = s_t(n)$  towards the peak of local density distribution. In this approach, the EM-like mean shift algorithm proposed in (Zivkovic and Kroese, 2004) is used to estimate the variance of local density distributions. A variance estimation allows the adaptation of the radius  $r$  corresponding to the current circular image features. Hence, taking into account movement in the depth of the camera, for tracking circular features in Hough space one has to incorporate an adaptation of radius  $r_t$ . Under the assumption that the distribution can be modeled as a Gaussian, parameters  $\hat{p}_n$  and  $\sigma_n$  are to be found which represent the center and variance of the distribution. These parameters should maximize following function:

$$f(\hat{p}_n, \sigma_n) = \operatorname{argmax}_{p, \sigma} \sum_{j=1}^M G(p_j; p_n; \sigma) I_H(p_j). \quad (5.15)$$

This can be solved iteratively by introducing a weight  $\lambda_j$  for each pixel  $p_j$  based calculating factors  $\lambda_j$  according to:

$$\lambda_j = \frac{G(p_j; \hat{p}_n; \sigma_n) I_H(p_j)}{\sum_{k=1}^M G(p_k; \hat{p}_n; \sigma_n) I_H(p_k)}. \quad (5.16)$$

Based on Eq. 5.16, an update rule can inferred as follows:

$$\hat{p}_n = \sum_{j=1}^M \lambda_j p_j, \quad (5.17)$$

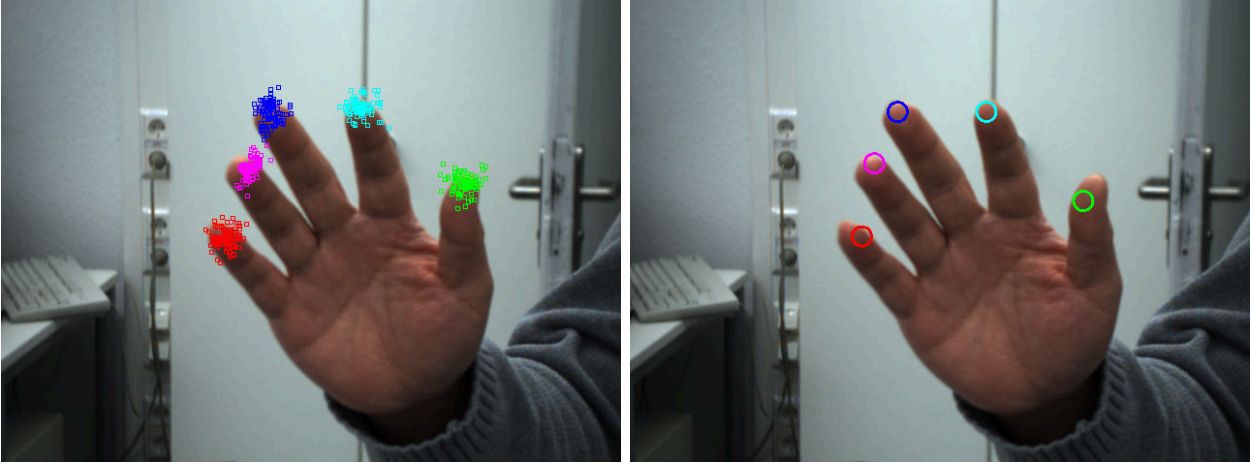


Figure 5.8.: Left: Fingertip estimates extracted from the particles. Right: Result image after the mean shift correction..

which shifts the center of the distribution towards its peak. A new estimate for the variance is obtained by evaluating following update term:

$$\sigma_n = c \sum_{j=1}^M \lambda_j (p_j - \hat{p}_n)(p_j - \hat{p}_n)^T, \quad (5.18)$$

where  $c$  is assumed to be a constant scaling factor. If convergence is achieved, the radius  $r$  is approximated by  $\sigma_n$ .

### Prediction

Providing a prediction of the movement of the objects to be tracked increases the robustness of a statistical tracking framework. For the fingertip tracking framework, we decided to train dynamical motion model in the form of a second-order auto-regressive (AR) process as proposed in (Blake and Isard, 2000), which is described as follows:

$$q_t - \bar{X} = A_1(q_{t-1} - \bar{q}) + A_2(q_{t-2} - \bar{q}) + b_0 \omega_k \quad (5.19)$$

where  $q_t \in \mathbb{R}^D$  denotes the current configuration,  $\bar{q}$  the mean configuration, and  $\omega_k \in [0, 1]$ . To learn the AR parameters  $A_1$ ,  $A_2 \in \mathbb{R}^{D \times D}$  and  $b_0 \in \mathbb{R}^D$ , training data is provided in the form of a configuration sequence  $Q = \{q'_0, \dots, q'_M\}$  whereby the sequence is generated by manual labeling of fingertips in each frame of a recorded image sequence.

Two AR models are trained to provide predictions for the fingertip movement for a nearly static hand pose as well as the movement of the hand itself. Based on the assumption that the motion of each finger is influenced by the motion of the neighbored fingers, the first model is trained with training data whose instances  $q'_i \in Q$  with  $D = N$  consists of the length of vector  $v_t^{j,j+1} = p_t^{j+1 \bmod N} - p_t^j$  between the fingertips  $j$  and  $j+1 \bmod N$ :

$$q'_i(j) = \left\| v_t^{j,j+1} \right\| \quad j = 1, \dots, N. \quad (5.20)$$

For finger  $j$ , this leads to following displacement vector:

$$\bar{v}_t(j) = \frac{q_t^j - q_{t-1}^j}{q_{t-1}^j} v_t^{j,j+1} - \frac{q_t^{j-1} - q_{t-1}^{j-1}}{q_{t-1}^{j-1}} v_t^{j-1,j}. \quad (5.21)$$

The second model which considers the global movement of the average position of all fingertips  $p_{mean}$  trained with data set formed of  $q'_i = p_{mean}$  with  $D = 2$  resulting into a overall displacement:

$$\hat{v}_t(j) = q'_i + \bar{v}_t(j). \quad (5.22)$$

Due to the coupled fingertip movements, the models behave well resulting in reasonable prediction of the finger displacements which supports the state estimation in the ensuing tracking procedure.

#### 5.4. Grasp Motion Data

The human capture methods described in Section 5.2 and Section 5.3 enable the acquisition of positional information about the hand and finger movements during grasping. For the instantiation of the grasp representation the captured trajectories have to be processed in order to form the data matrices  $\mathbf{X}_H$  and  $\mathbf{X}_F$  which describe the positions of the hand and the fingertips in the task space. While  $\mathbf{X}_H$  contains the positions of the grasping hand defined in an ego-centric coordinate system within the the grasping agent,  $\mathbf{X}_F$  is composed of fingertip trajectories locally defined with regard to the hand coordinate system. In this context, each motion capture method, markerless or marker-based, provides different information and, thus, depending on the used method the captured data has to be processed differently.

In case of marker-based captured motion data, considering the right hand to be the grasping hand, its position is denoted by the marker *RPM* which is placed on the back of the hand. To obtain the orientation of the hand, the markers *RPTS* and *RPPS* placed on each side of the hand are used to create a plane indicating the pose of the hand coordinate system. Therefore, the axis  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$  of the hand coordinate system are calculated as follows:

$$\begin{aligned} \mathbf{z} &= \frac{(\mathbf{x}_{RPM} - \mathbf{x}_{RMI})}{\|\mathbf{x}_{RPM} - \mathbf{x}_{RMI}\|} \\ \mathbf{y} &= \mathbf{z} \times \frac{(\mathbf{x}_{RPPS} - \mathbf{x}_{RPTS})}{\|\mathbf{x}_{RPPS} - \mathbf{x}_{RPTS}\|} \\ \mathbf{x} &= \mathbf{z} \times \mathbf{y}. \end{aligned} \quad (5.23)$$

In case of markerless captured motion data, it is assumed that the position of the grasping hand is denoted by the blob obtained by tracking the hand. The orientation of the hand is difficult to calculate due to the missing information. Thus, the orientation is approximated based on the captured fingertip positions. The plane based on which the coordinate frame is determined is spanned by the thumb, the middle finger, and the pinkie. Thus, the axis  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$  are determined by:

$$\begin{aligned} \mathbf{x} &= \frac{(\mathbf{x}_{Middle} - \mathbf{x}_{Thumb})}{\|\mathbf{x}_{Middle} - \mathbf{x}_{Thumb}\|} \times \frac{(\mathbf{x}_{Pinkie} - \mathbf{x}_{Thumb})}{\|\mathbf{x}_{Pinkie} - \mathbf{x}_{Thumb}\|} \\ \mathbf{y} &= \frac{(\mathbf{x}_{Middle} - \mathbf{x}_{Palm})}{\|\mathbf{x}_{Middle} - \mathbf{x}_{Palm}\|} \times \mathbf{x} \\ \mathbf{z} &= \mathbf{y} \times \mathbf{x}. \end{aligned} \quad (5.24)$$

For each frame, a grasp example in the hand coordinate system is calculated. Based on the matrix which describes the transformation from a base coordinate system (for markerless and marker-based data this base coordinate system is placed on the hip of the human subject) to the local hand coordinate system, the fingertip positions are defined as in-hand trajectories.

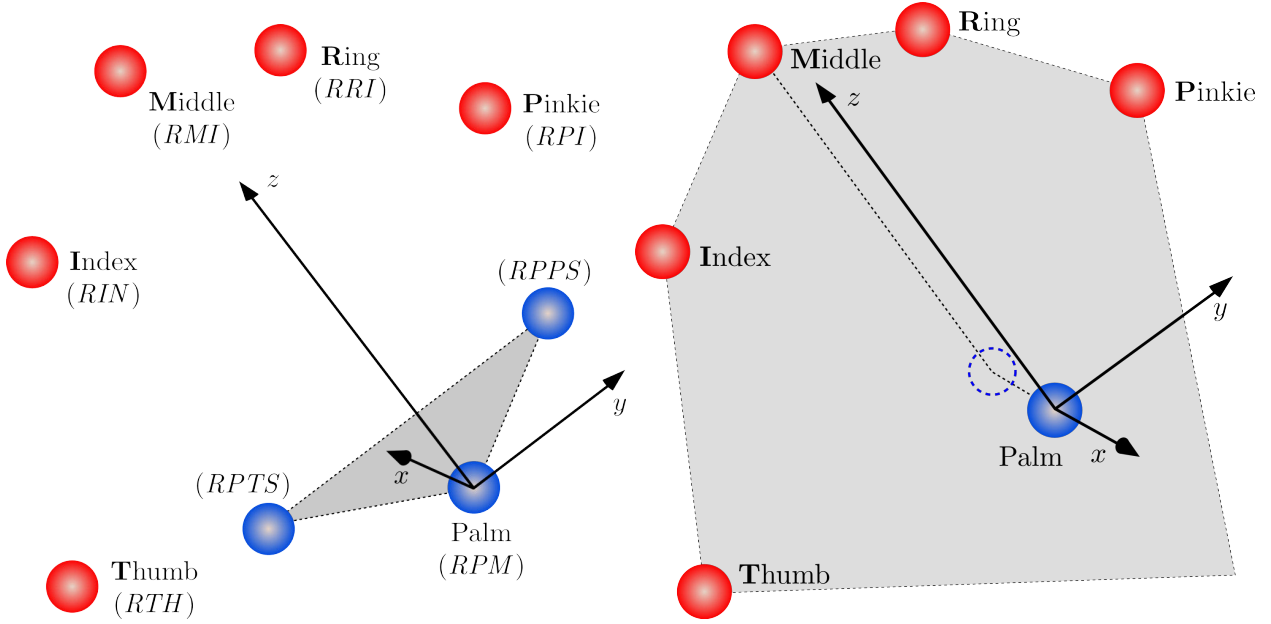


Figure 5.9.: Calculation of the local hand coordinate system based on the palm and fingertip positions captured with marker-based (left) and markerless (right) methods.

#### 5.4.1. Segmentation of Grasp Examples

**Marker-based** A segmentation is performed on the fingertip trajectories in order to cut off sequences which do not belong to the actual grasping movement and further to identify the time stamps which denote the end of the preshape and the enclose stage. Due to the high accuracy and frequency with which human motion can be captured with this system, segmentation of the data based velocity crossings proves to be sufficient. The end point of the enclose phase is defined as follows:

$$T_{end} = \operatorname{argmin}_t \sqrt{\sum_{i=0}^{N-1} \dot{G}(i)^2}. \quad (5.25)$$

**Markerless** Regarding the segmentation of the prehensile finger movement, due to increased noise, in addition to the rule stated in Eq. 5.25 the procedure is controlled by the hand-object relation as well. Hence, in this case,  $T_{end}$  is specified according following rule:

$$T_{end} = \operatorname{argmin}_t \sqrt{\sum_{i=0}^{N-1} \dot{G}(i)^2 + \frac{1}{N} \sum_{i=0}^{N-1} \|\mathbf{x}_g - G(i)\|}. \quad (5.26)$$

Regardless of which method is used for motion capture, in a further step, the point  $T_{pre}$  with  $0 < T_{start} < T_{pre} < T_{end}$  denoting the end of the preshape and the beginning of the enclose phase has to be determined. To do so,  $T_{pre}$  is defined as the point where the grasp aperture becomes maximal. Based on the assumption that the grip aperture is spanned between two virtual fingers  $T_{pre}$  is computed according to:

$$T_{pre} = \operatorname{argmax}_t \left\| G(0) - \frac{1}{N-1} \sum_{i=1}^{N-1} G(i) \right\|. \quad (5.27)$$

The movement segments are normalized using the Master Motor Map framework which is described in Section 7.3.1.

## 5.5. Summary

In this chapter, methods are presented which allow the observation of human grasp demonstrations. For the acquisition of accurate motion data which is captured at a high frame rate, a marker-based system is employed. The outcome of this system is considered as reference data based on which basic grasp primitives are learned in an offline manner using the representations described in Chapter 4. To enable a robot to observe human grasp demonstration invariant of time and location, an existing markerless upper body tracking framework has been extended by methods which allow the capturing of fine granular prehensile finger movements. Using a Hough transformation and a combination of particle filter and mean shift tracking, circular features representing the fingertips are localized and tracked. The extended framework uses the onboard systems of a robot, a pair of stereo cameras which provide peripheral and foveal views of the scene. Both methods, the upper body using peripheral and fingertip tracking using foveal vision, are coupled via a hand tracking procedure which provide foveal views of the hand. Regardless of marker-based or markerless, the captured motion is segmented and transformed into a local hand coordinate system in order to generate the grasp data needed for the instantiation of the proposed grasp representation.



## 6. Parameter Estimation for a Grasp Representation

The proposed Virtual Spring Grasp (VSG) representation as described in Section 4.3 incorporates a dynamical model which is parameterized to adopt an arbitrary grasping behavior. The model parameters, namely the spring constants, have to be adjusted in a way that the model fits exemplary observations of the grasp type to be represented. These observations are given by the grasp examples which have been recorded using the human motion capture methods described in Chapter 5. In the following, in Section 6.1, approaches for the estimation of parameters in dynamical systems will be briefly discussed. Subsequently, the concrete parameter estimation problem for the instantiation of the VSG representation is outlined in Section 6.2 and a method for the solution of the problem is presented in Section 6.3. Finally, a common method for the learning of the DMP representation for encoding the hand approach movement is reviewed in Section 6.5.

### 6.1. Parameter Estimation

The estimation of parameters in dynamical systems has been extensively studied in the area of system identification (e.g. in (Ljung, 1999) and (Young, 1981)). As stated in (Wu et al., 2010), parameter estimation methods can be categorized in online and offline methods whereas online methods are commonly applied in order to adapt a system's behavior to current observations. Especially for the design of adaptive control algorithms, the online estimation of model parameters plays a crucial role and has attracted much attention in the field of robotics. In this context, various works such as (Flacco and Luca, 2011), (Pham, 2001), and (Erickson et al., 2003) implemented an adaptive control mechanism using a recursive least squares approach to accommodate current measurements. An alternative approach based on a maximum-likelihood estimation method is introduced in (Swevers et al., 1997) which adapts robot model parameters to actual torque and joint angle measurement errors for the prediction of torque values in a dynamic setting. In (Gautier and Poignet, 2001) and (Sujan and Dubowsky, 2003), a technique based on Kalman Filtering is used for parameter adaptation. Online methods are suitable for implementing a flexible behavior into a system which varies over time. Since the parameter estimation does not rely on observations in the past, the response of the instantiated system is only considered to be valid within a small time window subject to adaptation.

In order to derive a global control law which is valid throughout time, a universal model that conforms with all available observations has to be generated. This requires that parameters have to be estimated in an offline manner. In this context, various approaches such as (Serban and Freeman, 2001) and (Ninness and Gibson, 2001) based on the least squares algorithm have been proposed for the estimation and instantiation of a given model structure. Approaches focused on the estimation of static parameters of mass spring damper have been introduced in (Becedas et al., 2009), (Lloyd et al., 2007), and (Majjad, 1997) using a recursive weighted least squares algorithm. However, most of these approaches are based on the assumption that the model to be parameterized is capable of reproducing the exact behavior which is observed in experimental measurements. Without prior knowledge and with sufficiently large data sets, offline methods can provide globally feasible parameter estimations by minimizing the error between model output and the corresponding output observations. In this thesis,

the representation of a grasp relies on a strongly simplified model of the hand. Hence, in the following, an estimation scheme is proposed which is tailored for the instantiation of the proposed VSG representation under consideration that input as well as output observations are contaminated with noise.

## 6.2. Problem Statement

To induce a certain grasp typical behavior into the VSG representation, the model parameters  $\mathbf{K} \in \mathbb{R}^{N \times N}$  and  $\mathbf{k}_c, \mathbf{k}_s \in \mathbb{R}^2$  have to be specified accordingly. Based on observations which exemplarily describe the behavior to be reproduced, it is desired to determine model parameters which minimize the error between the model output and the observed output.

### 6.2.1. Observational Data

Based on the trajectory  $\mathbf{G} := (\mathbf{X}_H, \mathbf{X}_F)$  defined in Eq. 4.1 and Eq. 4.2, the input and output observations are derived. These observations form the data used to instantiate a VSG representation that involves  $N$  fingers and set of finger springs  $\{s_1, \dots, s_M\}$ . To describe the data in a compact way, a lookup table  $I : (\{1, \dots, N\}, \{1, \dots, M\}) \rightarrow \{1, \dots, N\}$  is introduced indicating which mass bodies are connected via a certain spring. Hence, for a query with body  $b_i$  and a spring  $s_m$  placed between  $b_{j_0}$  and  $b_{j_1}$  the mapping  $I$  is defined as follows:

$$I(n, m) = \begin{cases} j_0 & \text{if } j_1 = i \\ j_1 & \text{if } j_0 = i \\ i & \text{else} \end{cases} \quad (6.1)$$

With  $I$ , input observations are created in the form of data matrices  $\mathbf{X}_{ss}, \mathbf{X}_{cs} \in \mathbb{R}^{3N \times T}$  and  $\mathbf{X}_{fs} \in \mathbb{R}^{3N \times TM}$ . The current displacements of the virtual finger springs at the current time frame  $0 < t \leq T$  are described by  $\mathbf{X}_{fs}(t) \in \mathbb{R}^{3N \times M}$  which forms  $\mathbf{X}_{fs} = (\mathbf{X}_{fs}(1) \times \dots \times \mathbf{X}_{fs}(t))$ . For a finger  $i$ , the entries of the data matrix  $\mathbf{X}_{fs}(t)$  are defined as follows:

$$\mathbf{X}_{fs}(t; n, m) = (\mathbf{X}_F(n, t) - \mathbf{X}_F(I(i, m), t)) \frac{(d_{i,I(i,m)} - l_m)}{d_{i,I(i,m)}} \quad (6.2)$$

where  $(i-1) \cdot 3 \leq n < i \cdot 3 \cdot N$  and  $0 < j \leq M$ . The spring  $m$  is described by the current length  $d_{i,I(i,m)}$  and the equilibrium length  $l_m$ . The displacements concerning the stabilization and the contact springs is represented by  $\mathbf{X}_{ss}$  and  $\mathbf{X}_{cs}$ . For the entire time series, based on the assumption that the contact configuration is equivalent to the final fingertip configuration at time  $T$ , the entries of the data matrices  $\mathbf{X}_{ss}$  and  $\mathbf{X}_{cs}$  are defined as follows:

$$\mathbf{X}_{cs}(n, t) = (\mathbf{X}_F(n, T) - \mathbf{X}_F(n, t)) \frac{(d_{c_i} - l_{c_i})}{d_{c_i}} \quad (6.3)$$

$$\mathbf{X}_{ss}(n, t) = \mathbf{X}_F(n, t) \frac{(d_{s_i} - l_{s_i})}{d_{s_i}}. \quad (6.4)$$

The output observations are represented by a data matrix  $\mathbf{X}_o \in \mathbb{R}^{3N \times T}$  with:

$$\mathbf{X}_o(n, t) = \ddot{\mathbf{X}}(n, t) \quad (6.5)$$

which basically describes the acceleration profile of the fingertips. To complete the system of differential equations as described in Eq. 4.26, the velocities of the fingertips as well as the



external forces acting on the corresponding mass bodies have to be specified. The relative velocities are combined in the matrix  $\mathbf{X}_v$ :

$$\mathbf{X}_v(t) = \frac{1}{\Delta t} [\dot{\mathbf{X}}_{fs}(t); \dot{\mathbf{X}}_{cs}(t); \dot{\mathbf{X}}_{ss}(t)] \quad (6.6)$$

with

$$\dot{\mathbf{X}}_{fs} = \mathbf{X}_{fs}(t) - \mathbf{X}_{fs}(t-1), \dot{\mathbf{X}}_{cs} = \mathbf{X}_{cs}(t) - \mathbf{X}_{cs}(t-1), \dot{\mathbf{X}}_{ss} = \mathbf{X}_{ss}(t) - \mathbf{X}_{ss}(t-1)$$

and  $\Delta t$  denoting the time interval which lies between the measurements.

To simplify the description of the problem and the corresponding solution, the entries of the spring constant matrix  $\mathbf{K}$  are rearranged into the form of a vector  $\mathbf{k}_f \in \mathbb{R}^M$ . Therefore, the parameter estimation problem is solved by determining the stiffness parameters  $\mathbf{k} = [\mathbf{k}_f; \mathbf{k}_c; \mathbf{k}_s]$  with  $\mathbf{k}_f \in \mathbb{R}^M$ ,  $\mathbf{k}_c \in \mathbb{R}^2$ , and  $\mathbf{k}_s \in \mathbb{R}^2$  which satisfy following equation:

$$\mathbf{X}_o(t) = \mathbf{X}(t)\mathbf{k} + \mathbf{X}_v(t)\boldsymbol{\zeta}, \quad (6.7)$$

with the damping parameter  $\boldsymbol{\zeta} = [\boldsymbol{\zeta}_f; \boldsymbol{\zeta}_c; \boldsymbol{\zeta}_s]$  and  $\mathbf{X}(t) = [\mathbf{X}_{fs}(t); \mathbf{X}_{cs}(t); \mathbf{X}_{ss}(t)]$  which subsumes the displacement matrices defined in Eq. 6.2 and Eq. 6.4. The problem stated in Eq. 6.7 is referred to as a multivariate nonlinear least squares problem. The nonlinearity is induced by the damping term  $\mathbf{X}_v(t)\boldsymbol{\zeta}$  which, due to the assumption that the mass spring damper systems are critically-damped, is defined as a function of the corresponding spring stiffness. To solve a nonlinear least squares problem, one has to minimize the residuals between the observations and the response of the parameterized model. Hence, based on Eq. 6.7 an optimization problem of the following form can be formulated:

$$\begin{aligned} \min f_E(\mathbf{k}) &= \sum_{t=1}^T \|\mathbf{X}(t)\mathbf{k} + \mathbf{X}_v(t)\boldsymbol{\zeta} - \mathbf{X}_o(t)\| \\ &\text{subject to } \mathbf{k} \geq 0 \end{aligned} \quad (6.8)$$

As stated in Section 4.2, the inequality constraint ensures the nonnegativity of the solution, and, thus, the feasibility of the spring constant estimate. Due to the nonlinearity of the problem a closed-form solution does not exist. Hence, numerical methods have to be applied which starting from an initial solution provide a refinement towards a better solution in an iterative manner. As depicted in Figure 6.3, the obtained solution is not guaranteed to be optimal due to the existence of local minima. A further critical issue provoked by the damping term is the introduction of noise onto both sides of Eq. 6.7. Since input as well as output observations are contaminated with noise, even an optimal solution to Eq. 6.9 does not necessarily induce the desired behavior in the given model.

Therefore, for the instantiation of the VSG representation, an estimation scheme is proposed which combines various methods in order to efficiently identify and approximate the spring constant parameters from noisy data.

### 6.3. Parameter Estimation Scheme

In the following, to obtain spring constant estimates for  $\mathbf{k} \in \mathbb{R}^{M+4}$ , an iterative parameter estimation scheme has been developed. The least squares problem specified in Eq. 6.7 is split up in three separate problems since the estimation of each parameter depends on disjunct observations. A further aspect which supports this practice is that variations of each parameter can have very different impact on the behavior of the dynamical system. The decomposed solution of the problem enables an individual adaptation of the estimation method by specifying crucial estimation parameters relative to the model parameter. For the estimation of a single

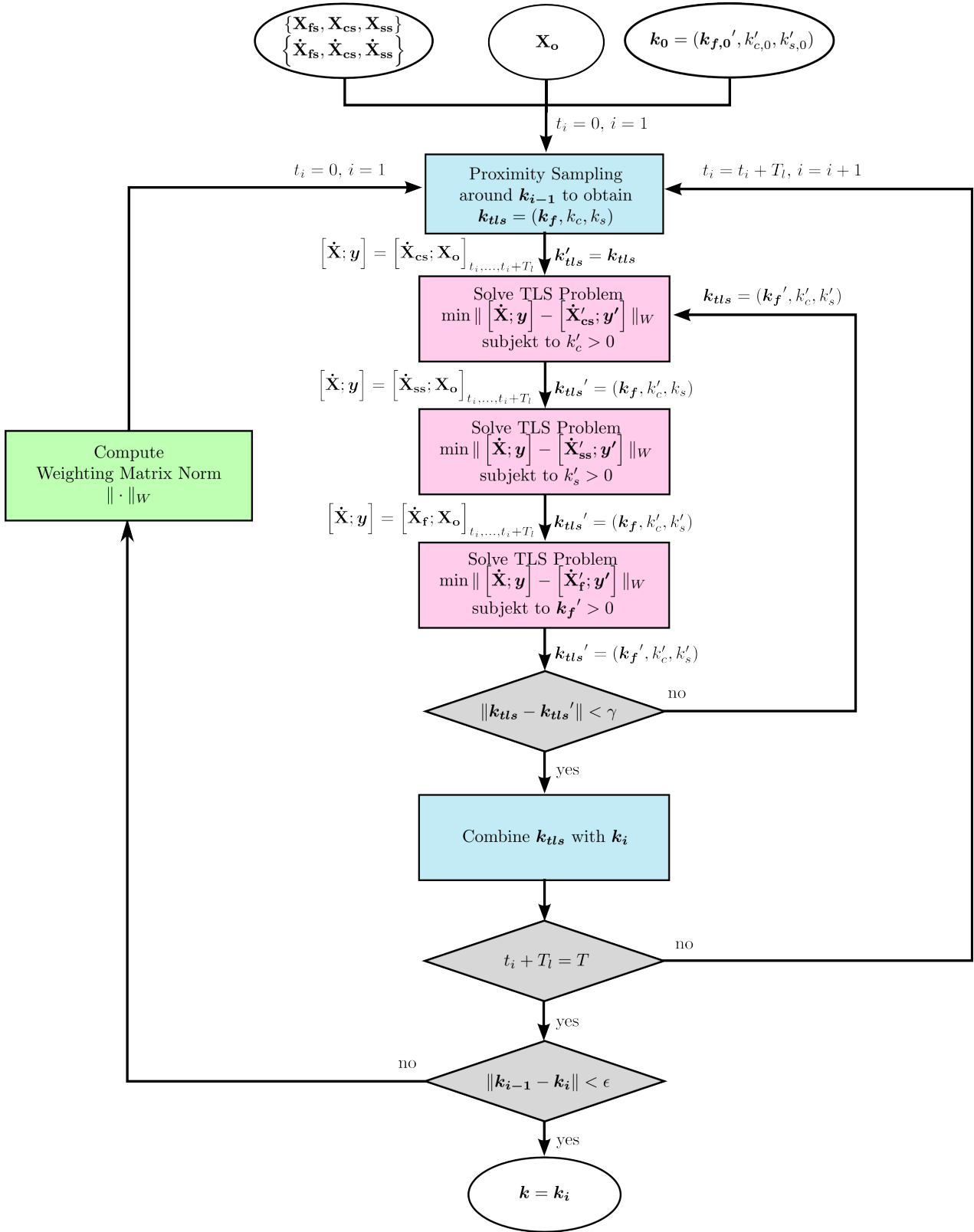


Figure 6.1.: Flowchart of the proposed estimation scheme.

parameter, it is assumed that the remaining parameters are known. Thus, for the estimation of  $\mathbf{k}_f$ , based on given solutions for  $\mathbf{k}_c$  and  $\mathbf{k}_p$  the corresponding optimization problem takes the following form:

$$\begin{aligned} \min f_{E_f}(\mathbf{k}_f) &= \sum_{t=1}^T \left\| \mathbf{X}_{fs}(t)\mathbf{k}_f - \dot{\mathbf{X}}_{fs}(t)2\sqrt{\mathbf{k}_f} - \mathbf{y}(t) \right\| \\ &\text{subject to } \mathbf{k}_f \geq 0 \end{aligned} \quad (6.9)$$

where  $\mathbf{y}(t) = \mathbf{X}_o(t) + \mathbf{X}_{ss}(t)\mathbf{k}_s - \dot{\mathbf{X}}_{ss}(t)2\sqrt{\mathbf{k}_s} + \mathbf{X}_{cs}(t)\mathbf{k}_c - \dot{\mathbf{X}}_{cs}(t)2\sqrt{\mathbf{k}_c}$ . Using a Total Least Squares (TLS) method (see Section 6.3), each iteration starts with the computation of spring constant estimates which satisfies Eq. 6.10 subject to the nonnegativity constraints (see Section 6.3.4). To increase the robustness and the efficiency of the parameter estimation, the proposed scheme uses a batch adaptation approach which divides the entire data set in smaller subsets. Each subset contains a smaller amount of data and, thus, allows the derivation of parameter estimates with less computational effort. However, the resulting estimates are considered as local solutions which become less likely to be valid for the entire data set with an increasing number of subsets. Therefore, the solutions have to be fused into a single estimate which is done by discarding infeasible solutions and combining feasible ones.

In a subsequent step, as described in Section 6.3.1, a weighting matrix is derived by evaluating the estimate on the given model. Based on the weighting matrix, a norm is defined which is used by the TLS method in upcoming iterations in order to accommodate individually scaled and correlated noise within the data. The estimation procedure converges after  $i$  iterations towards an estimation  $\hat{\mathbf{k}}'_i$  for the model parameters  $\mathbf{k}$ , when following conditions are satisfied:

$$\left\| \hat{\mathbf{k}}'_i - \hat{\mathbf{k}}'_{i-1} \right\| < \varepsilon \quad (6.10)$$

$$f_{E_f}(\mathbf{k}_f) < \delta_f \quad (6.11)$$

$$f_{E_c}(\mathbf{k}_c) < \delta_c \quad (6.12)$$

$$f_{E_s}(\mathbf{k}_s) < \delta_s \quad (6.13)$$

with  $\varepsilon$  being the convergence threshold and  $\delta_f$ ,  $\delta_c$ , and  $\delta_s$  denoting the upper error bounds for the residual functions  $f_{E_f}$ ,  $f_{E_c}$ , and  $f_{E_s}$ .

In this paragraph, the Total Least Squares (TLS) method is explained which is used for solving the problem stated in Eq. 6.10. The described procedure is applied for the determination of multivariate estimates for  $\mathbf{k}_f$ ,  $\mathbf{k}_c$  and  $\mathbf{k}_s$ . The TLS method is applied for the estimation of  $\mathbf{k}_f$ . For the sake of simplicity, for now, the nonnegativity constraints are disregarded. To obtain a local estimate for  $\mathbf{k}_f$  the data matrices  $\mathbf{X} = (\mathbf{X}_{fs}(t_j) \times \dots \times \mathbf{X}_{fs}(t_j + T_l))$ ,  $\dot{\mathbf{X}} = (\dot{\mathbf{X}}_{fs}(t_j) \times \dots \times \dot{\mathbf{X}}_{fs}(t_j + T_l))$  and the observation vector  $\mathbf{y} = (\mathbf{X}_o(t_i) \times \dots \times \mathbf{X}_o(t_j + T_l))$  are introduced which consider a series of  $T_l$  observations starting from  $t_j$ . To simplify the algorithmic description,  $\mathbf{k}' := \mathbf{k}_f$  is substituted. With  $\mathbf{X}$  and  $\dot{\mathbf{X}}$  Eq. 6.10 is reduced to a local residual functional:

$$f_E(\mathbf{k}) = \mathbf{X}\mathbf{k}' - \dot{\mathbf{X}}2\sqrt{\mathbf{k}'} - \mathbf{y}. \quad (6.14)$$

By introducing  $\mathbf{y}' = \mathbf{X}\mathbf{k}' - \dot{\mathbf{X}}2\sqrt{\mathbf{k}'}$  which denotes the response of the model parameterized with  $\mathbf{k}'$ , the unconstrained least squares formulation in Eq. 6.14 can be written as a quadratic optimization problem:

$$\mathbf{k}'_{ls} = \underset{\mathbf{k}'}{\operatorname{argmin}} \left\| \mathbf{y} - \mathbf{y}' \right\|_2. \quad (6.15)$$

Deriving the normal equations of Eq. 6.15 leads to an overdetermined system with  $3 \cdot N > M$  equations based on which an estimate can be obtained with common least squares methods that solves to  $\mathbf{X}\mathbf{k}'_{ls} - \dot{\mathbf{X}}2\sqrt{\mathbf{k}'_{ls}} = \mathbf{y}'$  and minimizes Eq. 6.14. This applies only if the model

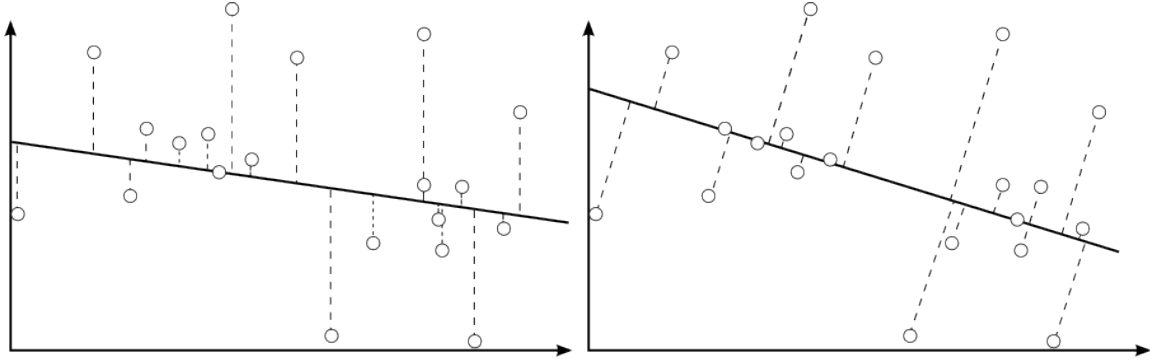


Figure 6.2.: Solution for a 2D functional regression problem obtained by applying the least squares solver (left) and the TLS solver (right).

response is allowed to be defined as  $\mathbf{y}' = \mathbf{y} + \Delta\mathbf{y}$  which is the corrected system of equations using a minimized correction term  $\Delta\mathbf{y}$  in order to accommodate for eventual noise. The input observations  $\mathbf{X}$  are assumed to be free of noise, and, thus, reflect the exact states of the given model. In most scenarios, this prerequisite does not hold.

In the case of the VSG representation, a grasp behavior is defined for a construct which strongly simplifies the structure of the grasping hand. Hence, due to modeling and measurement errors, the contamination of input observations with noise is inevitable. To attain an estimate which accounts for this circumstance, the optimization problem defined in Eq. 6.15 is extended in the following form:

$$\begin{aligned} \mathbf{k}_{tls} &= \operatorname{argmin}_{\mathbf{k}'} \left\| [\dot{\mathbf{X}}; \mathbf{y}] - [\dot{\mathbf{X}}'; \mathbf{y}'] \right\|_F \\ &\text{subject to } \mathbf{X}\mathbf{k}_{tls} - \dot{\mathbf{X}}'2\sqrt{\mathbf{k}_{tls}} = \mathbf{y}' \end{aligned} \quad (6.16)$$

with  $[\dot{\mathbf{X}}; \mathbf{y}] \in \mathbb{R}^{3N+1 \times M}$ ,  $\dot{\mathbf{X}}' = \dot{\mathbf{X}} + \Delta\dot{\mathbf{X}}$ , and  $\|\cdot\|_F$  denoting the Frobenius norm. According to (Golub and Loan, 1983) the problem stated in Eq. 6.17 is referred to as a Total-Least-Squares (TLS) problem with  $\mathbf{k}_{tls}$  being the corresponding TLS estimate.

### 6.3.1. Solving the TLS Problem

To derive a solution for Eq. 6.17 as proposed in (Markovsky and Huffel, 2007) a two-step iterative algorithm based on an Expectation-Maximization (EM) approach is applied. Each iteration starts with the estimation of  $\mathbf{k}_{tls}$  with fixed  $\mathbf{X}'$  which satisfies following term (E-step):

$$\mathbf{k}_{tls} = \operatorname{argmin}_{\mathbf{k}'} \left\| [\dot{\mathbf{X}}\mathbf{y}] - [\dot{\mathbf{X}}'\mathbf{y}_{k'}] \right\|_F. \quad (6.17)$$

In the subsequent step, data set is optimized to attain better fit with regard to estimation  $\mathbf{k}'$  which can be described as follows (M-step):

$$\mathbf{X}_{tls} = \operatorname{argmin}_{\mathbf{X}'} \left\| [\mathbf{X}\mathbf{y}] - [\tilde{\mathbf{X}}'\mathbf{y}_{k_{tls}}] \right\|_2. \quad (6.18)$$

To increase the chances of finding parameter estimates which provide a good fit, initial values have to be determined which indicate where to start the parameter optimization. The initialization procedure is implemented for the parameter estimation scheme is described in Section 6.3.2.

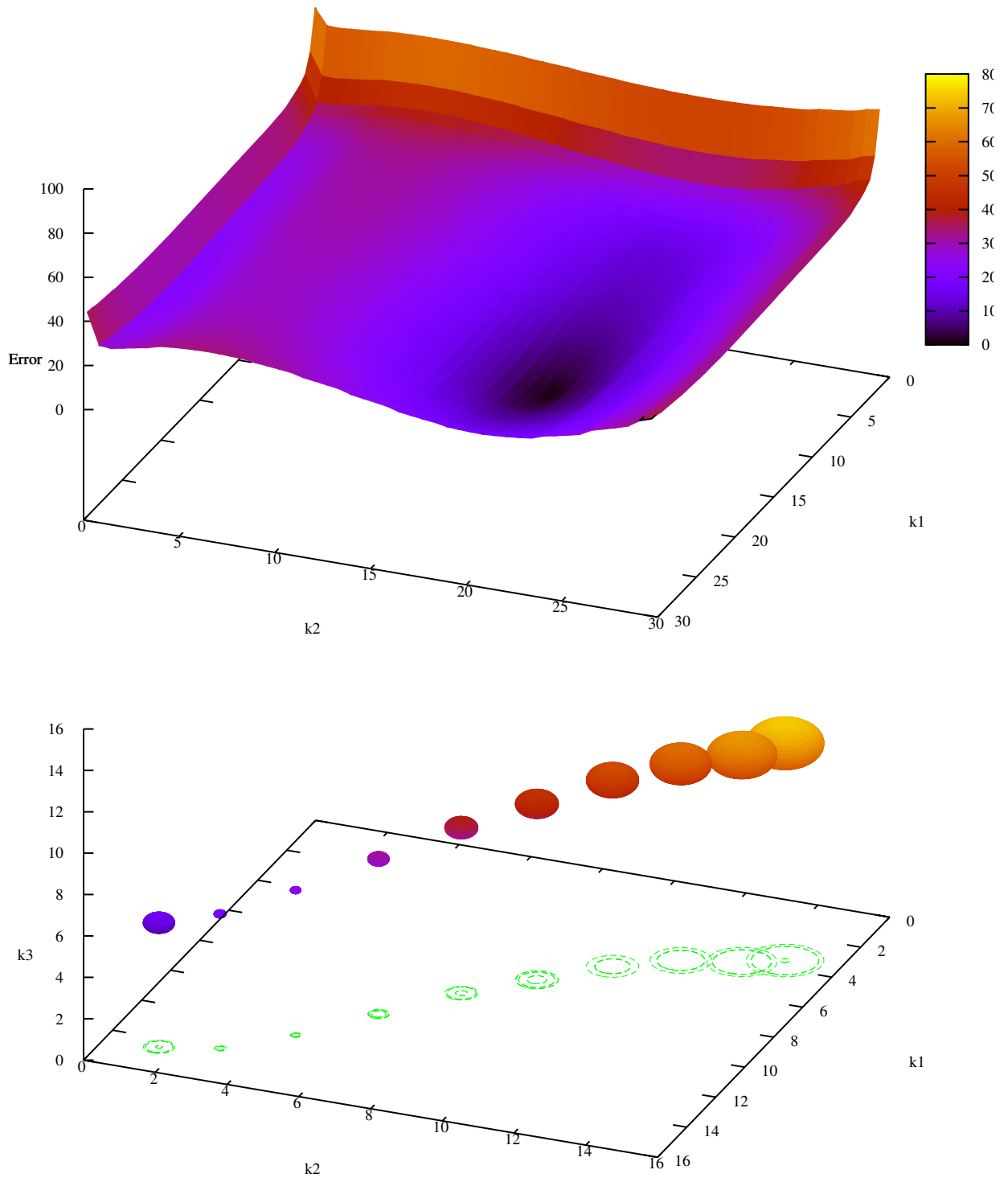


Figure 6.3.: Top: Mean squared error for the optimization of two spring constants. Bottom: Local minima for the optimization of three spring constants. Each minimum is represented as sphere where its radius represent the extend of the mean squared error.

### 6.3.2. Initial Solution

Inspired by (Andrieu et al., 2005), to diminish the risk of being stuck in local minima, a sequential Monte Carlo method is applied to sample the space in the proximity of a previous estimation  $\mathbf{k}_{i-1}$ . In order to generate a sample  $\hat{\mathbf{k}}_i$ , Gaussian random noise in the form of  $\boldsymbol{\psi}$  is added to the  $\mathbf{k}_{i-1}$ . Since the parameter estimation problem is solved separately for  $\mathbf{k}_f$ ,  $\mathbf{k}_c$ , and  $\mathbf{k}_s$  three different samples are generated as follows:

$$\begin{aligned}\hat{\mathbf{k}}_f &= \mathbf{k}_{i-1} + a_f \boldsymbol{\psi} \\ \hat{\mathbf{k}}_c &= \mathbf{k}_{i-1} + a_c \boldsymbol{\psi} \\ \hat{\mathbf{k}}_s &= \mathbf{k}_{i-1} + a_s \boldsymbol{\psi} \quad ,\end{aligned}\tag{6.19}$$

where  $a_f$ ,  $a_c$ , and  $a_s$  denote the individual scaling factors. For each sample  $\hat{\mathbf{k}}_i$ , the quadratic error function in Eq. 6.22 is evaluated and the parameter estimate  $\mathbf{k}'$  is specified according to:

$$\mathbf{k}' = \underset{\hat{\mathbf{k}}_i}{\operatorname{argmin}} g_E(\mathbf{k}').\tag{6.20}$$

### 6.3.3. Local Estimation of Spring Constant Parameters

To obtain an estimate  $\mathbf{k}_{tIs}$  which satisfies Eq. 6.17, based on the residual function, a quadratic problem is specified as follows:

$$\min g_E(\mathbf{k}') = \|f_E(\mathbf{k}')\|_F^2\tag{6.21}$$

$$= \operatorname{Tr}(f_E(\mathbf{k}')^T f_E(\mathbf{k}')).\tag{6.22}$$

A stationary point at which Eq. 6.22 becomes minimal is determined using the first derivative of  $g_E(\mathbf{k}')$ :

$$\mathbf{g}'_E(\mathbf{k}') = 2\mathbf{J}_E^T f_E(\mathbf{k}')\tag{6.23}$$

with  $\mathbf{J}_E \in \mathbb{R}^{3NT_i \times M}$  denoting the Jacobian matrix that contains the gradients of  $f_E(\mathbf{k}')$  and is defined as follows:

$$\mathbf{J}_E = \begin{pmatrix} \frac{\delta f_E(\mathbf{k}';1)}{\delta \mathbf{k}'(1)} & \cdots & \frac{\delta f_E(\mathbf{k}';1)}{\delta \mathbf{k}'(M)} \\ \vdots & \ddots & \vdots \\ \frac{\delta f_E(\mathbf{k}';T_i)}{\delta \mathbf{k}'(1)} & \cdots & \frac{\delta f_E(\mathbf{k}';T_i)}{\delta \mathbf{k}'(M)} \end{pmatrix}.\tag{6.24}$$

In general, simple optimization techniques based on search algorithms such as the Simplex method or gradient-based steepest descent approach can already be applied to derive an estimate for  $\mathbf{k}_{tIs}$  based on the information provided in Eq. 6.24, Eq. 6.22 and Eq. 6.23. However, these techniques suffer from several deficiencies such as slow convergence, zigzagging or getting easily trapped in local minima. To reduce the risk of suffering from these effects, additional information given by the second derivative can be exploited in order to direct the optimization procedure. For  $g_E(\mathbf{k}')$ , the second derivative is derived by determining its Hessian Matrix  $\mathbf{G}_E$  which is written as follows:

$$\mathbf{G}_E = 2\mathbf{J}_E^T \mathbf{J}_E + 2 \sum_{t=1}^{T_i} f_E(\mathbf{k}';t) \mathbf{T}_E(t),\tag{6.25}$$

where  $\mathbf{T}_E(j) \in \mathbb{R}^{3NT_i \times M}$  represents the Hessian matrix of  $f_E(\mathbf{k}')$  in the following form:

$$\mathbf{T}_E(j) = \begin{pmatrix} \frac{\delta f_E(\mathbf{k}';1)}{\delta k(1)\delta k(1)} & \cdots & \frac{\delta f_E(\mathbf{k}';1)}{\delta k(1)\delta k(M)} \\ \vdots & \ddots & \vdots \\ \frac{\delta f_E(\mathbf{k}';T_i)}{\delta k(M)\delta k(1)} & \cdots & \frac{\delta f_E(\mathbf{k}';T_i)}{\delta k(M)\delta k(M)} \end{pmatrix}.\tag{6.26}$$

$\mathbf{G}_E$  plays a crucial role in determining an efficient update function for the least squares estimate since it provides a measure which relates the model response to the variations of parameters to be estimated. However, the computation and inversion of the Hessian matrix is computationally expensive which is why most numerical optimization techniques use an approximation of  $\mathbf{G}_E$  in order to derive an update step in an iterative scheme. An algorithm which proceeds this way and which is applied to solve Eq. 6.22 is the Levenberg-Marquardt method. The algorithm, which was first introduced in (Levenberg, 1944), combines the Gauss-Newton and the steepest descent method and, thus, unites the advantages of both methods. Based on the assumption that  $\mathbf{G}_E \approx (J_E^T)J_E$  an update equation for  $\mathbf{k}'$  is formulated which changes a previous solution proportional to a step size into a direction given by the gradient and scaled by the inverse Hessian:

$$\mathbf{k}'_{i+1} = \mathbf{k}'_i - b(J_{E,i}^T J_{E,i} + \mu_i I)^{-1} J_{E,i}^T f_E(\mathbf{k}'_i). \quad (6.27)$$

To avoid explicit inversion of  $J_{E,i}^T J_{E,i} + \mu_i I$ ,  $p_k$  is introduced to derive following system of equations:

$$(J_{E,i}^T J_{E,i} + \mu_i I) p_k = J_{E,i}^T f_E(\mathbf{k}'_i), \quad (6.28)$$

which can be efficiently solved using matrix factorization techniques such as the Cholesky or the QR method. Here, the QR method is preferred due to its numerical stability. To accommodate for an ill-conditioned Jacobian matrix the Levenberg-Marquardt algorithm incorporates a regularization term in the form  $\mu_i I$ . Furthermore, the regularization parameter  $\mu_i$  can be exploited to control the convergence behavior of the algorithm. When  $g_E(\mathbf{k}'_i)$  approaches a local minimum a smaller value is assigned to  $\mu_{i+1}$  to achieve faster convergence. If minimization of  $g(\mathbf{k}'_i)$  fails,  $\mu_{i+1}$  is set to a higher value, which slows down the convergence, and, thus, limits the divergence of the optimization. How to adjust the regularization parameter  $\mu$  has been studied extensively in various works. Based on (Yamashita and Fukushima, 2001), a simple and effective update rule is inferred for the problem at hand:

$$\mu_{i+1} = \mu_i(g_E(\mathbf{k}'_i)), \quad (6.29)$$

which allows quadratic convergence. Further details on the theory of the Levenberg-Marquardt method can be found in (Lourakis, 2005).

### 6.3.4. Nonnegativity Constraints

So far, a solution for the unconstrained parameter estimation problem has been proposed in the form of the TLS method. In this section, the inequality constraints ensuring the nonnegativity of the solution are incorporated in the estimation procedure. To guarantee the optimality when solving Eq. 6.9 subject to the nonnegativity constraints, the estimate has to satisfy the Karush-Kuhn-Tucker (KKT) conditions. Considering the KKT optimality conditions, Eq. 6.9 is extended as follows:

$$\min f_E(\mathbf{k}') = \sum_{t=1}^{T_l} \|\mathbf{X}(t)\mathbf{k}' + \mathbf{X}_v(t)\boldsymbol{\zeta} - \mathbf{X}_o(t)\| \quad (6.30)$$

$$\text{subject to} \quad \mathbf{k}' \geq \mathbf{0} \quad (6.31)$$

$$\dot{f}(\mathbf{k}') \geq \mathbf{0} \quad (6.32)$$

$$\dot{f}(\mathbf{k}')^T \mathbf{k}' \geq 0 \quad (6.33)$$

To solve nonnegative least squares problems, the estimation procedure follows the active set method proposed by (Lawson and Hanson, 1974). This algorithm identifies an active subset of constraints violating the KKT conditions and sets the corresponding regression coefficients

to zero. Hence, at the beginning of each iteration the passive constraint set is determined according following rule:

$$I_p = \{m | \mathbf{k}'(m) = 0, \dot{f}(\mathbf{k}'; m) > 0\}, \quad (6.34)$$

where  $I_p$  holds the indices of the spring parameters estimates which satisfy the KKT conditions. Accordingly, an active set  $I_a$  is defined which is equivalent to the complement to  $I_p$ . For single parameters in  $\mathbf{k}'$  which are indexed by entries of the passive set, the KKT conditions are fulfilled. In contrast, parameters belonging to the active set and for which  $\mathbf{k}'(m) = 0$  and  $\dot{f}(\mathbf{k}'; m) \leq 0$  holds, have to be optimized further. Hence, focusing on the optimization of the active set, a reduced least squares problem of the following form has to be solved:

$$\min f_{E_a}(\mathbf{k}') = \frac{1}{T_l} \sum_{t=t_i}^{T_l} \sqrt{\sum_{l=1}^{|I_a|} (\mathbf{k}'(I_a(l)) \mathbf{X}(I_a(l), t) - \mathbf{y}(I_a(l), t))^2} \quad (6.35)$$

$$\text{subject to } \mathbf{k}' \geq 0 \quad (6.36)$$

Based on  $\mathbf{k}'_a$  which solves Eq. 6.36, an updated estimate of the spring constant  $\mathbf{k}'$  is formed as follows:

$$\mathbf{k}'(m) = \begin{cases} 0 & , m \in I_p \\ \mathbf{k}'_a(m) & , m \in I_a \end{cases} \quad (6.37)$$

Eq. 6.36 is treated as an unconstrained least squares problem and a parameter estimate for the passive subset is attained by applying the previously described Levenberg-Marquardt algorithm. The active and passive subsets are iteratively reevaluated and modified in order to find the true active subset by pushing an active coefficient towards a feasible solution (in most cases towards zero) and simultaneously checking whether the KKT conditions are satisfied.

## Regularization

As previously mentioned, Eq. 6.9 states an ill-posed problem due to data noise and the model simplifications. Hence, the presented parameter estimation problem does not have a unique solution in general. Therefore, a regularization procedure in the form of the Tikhonov regularization is introduced to make the system uniquely solvable.

Given an approximation of the spring constants  $\mathbf{k}'$  an error estimation is needed to determine the quality of the approximation. For that reason, the error function  $g_E$  is extended by:

$$g_E(\mathbf{k}') = \|\mathbf{f}_E(\mathbf{k}')\|_F^2 + \gamma \|\mathbf{k}'\|, \quad (6.38)$$

giving a rating for an approximation of the spring constants  $\mathbf{k}'$ . The regularization parameter  $\gamma$  is estimated by analyzing simulated grasp trajectories. For that purpose, dynamical systems with known spring constants are considered and searched for the  $\gamma$  minimizing the difference between the real spring constants and their estimations.

### 6.3.5. Estimation of Data Correction

Subsequent to the estimation of the model parameters, a correction term for the underlying data is computed which allows modifying the data in such a way that it better fits the model description and, thus, might lead to a more accurate estimation. As previously mentioned, the noise regarding the input observations is introduced by the damping term. Hence, based on the assumption that  $\mathbf{k}_{ts}$  is an optimal solution to the system, a correction term for the velocity matrix  $\dot{\mathbf{X}}'$  is estimated by minimizing the following error functional:

$$\min g_M(\dot{\mathbf{X}}') = \|\mathbf{f}_M(\dot{\mathbf{X}}')\|_2^2 \quad (6.39)$$



with

$$f_M(\dot{\mathbf{X}}') = \mathbf{y}' - \mathbf{X}\mathbf{k}_{tls} - \dot{\mathbf{X}}'\zeta'. \quad (6.40)$$

Instead of estimating a correction term for the entire matrix  $\mathbf{X}$ , Eq. 6.40 is solved for each data point separately. Although  $f_M(\dot{\mathbf{X}}')$  is linear, the system of normal equations is underdetermined since  $n > m$ . Therefore, a unique optimal solution does not exist. However, there exists a unique minimizer with a minimum norm in terms of the Singular Value Decomposition of  $\dot{\mathbf{X}}'$ . A local minimum in the vicinity of the actual measurements  $\dot{\mathbf{X}}$  can be easily computed using following update rule:

$$\dot{\mathbf{X}}_{tls} = \dot{\mathbf{X}} - \mathbf{J}_M \mathbf{J}_M^T (f_M(\dot{\mathbf{X}})) \quad (6.41)$$

with  $\mathbf{J}_M = \mathbf{X}_o$ . Within a single iteration Eq. 6.41 converges to the corrected velocity matrix with which the  $\mathbf{X}' = \mathbf{X}_{tls}$ . Thus, Eq. 6.17 can be specified in a way that  $\mathbf{k}_{tls}$  solves the problem in iteration  $i$ .

## 6.4. Weight Formation

The TLS method is capable of dealing with errors in the observations which are independent variables with zero mean and are identically distributed. For measurement errors which are individually scaled and correlated between the data points the TLS method has to be extended. For this reason, a weight matrix is incorporated in the least squares method allowing to penalize outliers and individual frames. A more generalized formulation of the TLS problem is obtained by introducing a weighting matrix  $\mathbf{W} \in \mathbb{R}^{T_i \times T_i}$ . The Frobenius norm in Eq. 6.17 is extended by the weight matrix norm  $\|\cdot\|_W$  which is defined as follows:

$$\|\mathbf{X}\|_W := \sqrt{\text{Tr} \mathbf{X}^T \mathbf{W} \mathbf{X}}. \quad (6.42)$$

So far, the TLS problem has been solved under the assumption that  $\mathbf{W} = \mathbf{I}$  for which  $\|\cdot\|_W$  is equivalent to  $\|\cdot\|_F$ . Without any prior knowledge of the observational error, the only way to obtain information about the quality of the model fit lies in the analysis of the cross covariances which are computed as follows:

$$\text{cov}(t_u, t_v) = (\mathbf{X}_o(t_u) - \mathbf{X}(t_u)\mathbf{k} - \mathbf{X}_v(t_u)\zeta)(\mathbf{X}_o(t_v) - \mathbf{X}(t_v)\mathbf{k} - \mathbf{X}_v(t_v)\zeta)^T \quad (6.43)$$

with  $t_i < t_u, t_v < T_i$ . The covariances between all time stamps are subsumed in a covariance matrix  $\mathbf{C}_i \in \mathbb{R}^{T_i \times T_i}$ :

$$\mathbf{C}_i = \begin{pmatrix} \text{cov}(t_i, t_i) & \cdots & \text{cov}(t_i, T_L) \\ \vdots & \ddots & \vdots \\ \text{cov}(T_L, T_i) & \cdots & \text{cov}(T_L, T_L) \end{pmatrix}. \quad (6.44)$$

A better model fit is obtained when the variances for all parameters can be minimized. To obtain a minimum variance estimate for  $\mathbf{k}$ , the most suitable choice for the weight matrix  $\mathbf{W}$  is the inverse of the covariance matrix  $\mathbf{C}_i$ . Based on the assumption that the error covariance between different data points is zero and the variance of the error term does not remain constant in time,  $\mathbf{C}_i$  is transformed to  $\mathbf{C}_i^1$  according following rule:

$$\text{cov}(t_u, t_v) = \begin{cases} \sigma_{t_u} & \text{if } t_u = t_v \\ 0 & \text{if } t_u \neq t_v \end{cases} \quad (6.45)$$

which results in:

$$\mathbf{C}_i^1 = \begin{pmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_N^2 \end{pmatrix}. \quad (6.46)$$

For correlated errors in different time-related observations, it is assumed that the covariance of the error term across different time periods is no longer zero. Hence, considering heteroscedastic effects, an error covariance matrix  $\mathbf{C}_i^2$  can be defined as:

$$\mathbf{C}_i^2 = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1^2 & & & 0 \\ \rho\sigma_1^2 & \sigma_2^2 & \rho\sigma_2^2 & & \\ & \ddots & \ddots & \ddots & \\ & & \rho\sigma_{T-2}^2 & \sigma_{T-1}^2 & \rho\sigma_{T-1}^2 \\ 0 & & & \rho\sigma_{T-1}^2 & \rho\sigma_T^2 \end{pmatrix}, \quad (6.47)$$

where

$$\text{cov}(t_u, t_v) = \begin{cases} \sigma_{t_u} & \text{if } t_u = t_v \\ \rho\sigma_{\min(t_u, t_v)} & \text{if } |t_u - t_v| = 1 \\ 0 & \text{if } |t_u - t_v| > 1 \end{cases} \quad (6.48)$$

holds. Depending on the observations, the weighting matrix can be defined as  $\mathbf{W} = \mathbf{C}_i^{1-1}$  or  $\mathbf{W} = \mathbf{C}_i^{2-1}$ . In this thesis, the second form is chosen. Due to the simplified description of the grasp and the applied human observation techniques, an accurate modeling of the dynamics of these processes is not possible. The TLS method underestimates the true uncertainty of the parameters by addressing merely zero-mean error.

#### 6.4.1. Weighted Estimation of Spring Constant Parameters

For the incorporation of a weighting matrix  $\mathbf{W} \in \mathbb{R}^{T_i \times T_i}$  in the TLS method, the quadratic error functional introduced in Eq. 6.17 has to be adapted accordingly. Using the weight norm  $\|\cdot\|_W$ , the quadratic optimization problem transforms to:

$$\min_{\mathbf{k}'} g_W(\mathbf{k}') = \frac{1}{2} \|f_E(\mathbf{k}')\|_W^2 \quad (6.49)$$

$$= \frac{1}{2} f_E(\mathbf{k}')^T \mathbf{W} f_E(\mathbf{k}'). \quad (6.50)$$

Consequently, the corresponding gradient to the error functional is determined by:

$$\dot{g}_W(\mathbf{k}') = \mathbf{J}_E^T \mathbf{W} f_E(\mathbf{k}'), \quad (6.51)$$

where the second derivative is calculated as follows:

$$\ddot{g}_E(\mathbf{k}') = \mathbf{J}_E^T \mathbf{W} \mathbf{J}_E + \sum_{j=1}^{T_i} \mathbf{f}_{E, \mathbf{k}}(j) \mathbf{W} \mathbf{G}_E(j) \quad (6.52)$$

with  $\mathbf{J}_E \in \mathbb{R}^{T_i \times M}$  denoting the Jacobian which contains the gradients of  $f_E(\mathbf{k}')$ .

### 6.5. Instantiation of the Dynamic Movement Primitive

In order to encode a given hand approach movement  $\mathbf{X}_H$  in the form of a DMP, the transformation systems have to be adapted accordingly. The adaptation is accomplished by determining the weights  $\mathbf{w} \in \mathbb{R}^P$  for the nonlinear perturbation term  $f_H$  with which an attractor landscape is shaped corresponding to the demonstrated trajectory  $\mathbf{X}_H$ . For the temporal invariance

of the transformation system, following replacements  $\mathbf{v} = \frac{\dot{\mathbf{x}}}{\tau}$  and  $\dot{\mathbf{v}} = \frac{\ddot{\mathbf{x}}}{\tau}$  are carried out in Eq. 4.28. Rearranging the equation leads to:

$$f_H(s(t)) = \frac{\ddot{\mathbf{x}}(t)}{\tau^2} - k_H(\mathbf{x}_g - \mathbf{x}(t)) + \zeta_H \frac{\dot{\mathbf{x}}(t)}{\tau}. \quad (6.53)$$

Inserting  $\mathbf{X}_H$ ,  $\dot{\mathbf{X}}_H$ , and  $\ddot{\mathbf{X}}_H$  into Eq. 6.53 results in a perturbation force value  $f'_H(s(t))$ . Fitting these values to the perturbation representation in Eq. 4.29 can be solved as a regression problem for the weight vector  $\mathbf{w}$  which can be written as follows for each dimension:

$$\mathbf{w}_n = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{t=1}^{T_e} \sum_{p=1}^P (f'_H(s(t)) - \Psi_p(s(t))\mathbf{w})^2, \quad (6.54)$$

for  $i = 1, \dots, M + N$ . This problem can be solved efficiently by standard locally-linear regression methods. In this thesis, the Receptive Field Weighted Regression (RFWR) method as introduced in (Schaal and Atkeson, 1998) has been chosen. The proposed technique is specifically suitable for online learning problems since it follows an eager learning policy. The implemented strategy is based on an iterative update of the existing model concerning incoming data. The model is composed of a number of  $P$  locally linear models which are denoted as Receptive Fields and formalized as Gaussian kernel functions with which the relation between input and output observations is approximated. The influence of a new data point on a Receptive Field is determined based on the distance to its center and its variance. For an algorithmic description and further implementation details on the RFWR method the reader is referred to (Schaal and Atkeson, 1998).

## 6.6. Summary

In this chapter, an optimization scheme has been presented which allows the estimation of parameters needed for the instantiation of the proposed grasp representation. The proposed scheme is specifically designed to allow the efficient instantiation of the VSG representation from noisy data where the spring constant estimates are derived by solving a nonlinear optimization problem. Towards an efficient solution, the original problem is broken down to sub-problems which, through a batch processing approach, is solved for local subsets of observations. By introducing the Total Least Squares method for the spring constant estimation, it has been shown that the VSG Representation is based on a strong simplification of the grasping hand, and, thus, the grasping action. Therefore, one has to assume that input as well as output observations are error-prone due to this modeling error. To obtain physically feasible spring constant estimates, the TLS problem is solved subject to nonnegativity constraints. In order to accommodate an individually scaled and correlated non zero-mean error, a weight matrix norm is introduced for attaining a better fit of the data to the model. Concerning the hand approach movement, a regression technique in the form of the RFWR method is used to obtain the weights of local models which constitute the nonlinear perturbation term used for shaping the attractor landscape. Both, the VSG and the DMP representation, are defined in the task space and the instantiations of these representations are considered as separate procedures. Hence, grasp examples are divided in two different observation sets which describe the prehensile fingertip movements and the hand approach movement in the task space. The methods which are needed to acquire and to process grasp examples from human observation are subject of the following chapter.



## 7. Grasp Learning and Execution

In the following, a framework is proposed which incorporates the introduced grasp representation and corresponding methods needed to acquire grasp data and to instantiate the representation. The purpose of this framework is to enable a humanoid robot to learn different grasps from the observation of human demonstrations and to apply the acquired grasp knowledge under varying task and object constraints. First, in Section 7.1, similar frameworks are reviewed. Subsequently, the proposed framework is introduced in Section 7.2. A major focus of this chapter is set on additional methods which are necessary for the implementation of the desired grasp learning behavior. In Section 7.3, the correspondence problem between the human and the robotic embodiment is addressed. A central role in the mapping process is played by the Master Motor Map, framework which is subject of Section 7.3.1. Finally, several aspects regarding the reproduction of represented grasping actions are discussed in Section 7.5.

### 7.1. Related works

Motivated by the vision of endowing robots with cognitive abilities to observe, to generalize and to learn and, thus, enhancing the interaction between humans and robots, programming by demonstration (PbD) has become an attractive paradigm and a much discussed topic in the fields of robotics. With the development of humanoid robots the concept of imitation learning arose from the PbD paradigm in order to address the novel challenges. Going beyond the mainly engineering aspect of PbD, imitation learning approaches emulate human skill acquisition behavior by making use of findings on human physiology and psychology. Following methodology in PbD respectively imitation learning, numerous approaches focused on the observation, representation, and reproduction of coarse arm movements in manipulation and grasping actions have been introduced ((Schaal, 1997), (Pastor et al., 2009), (Kulić et al., 2008), (Calinon et al., 2007), (Hersch et al., 2008), (Ijspeert et al., 2013)) using motion data which mostly originate from marker-based systems or kinesthetic teaching.

In contrast to that, few works have addressed the learning of prehensile finger movements from human observation. In (Zoellner et al., 2004), framework is introduced using multiple stereo cameras and two data gloves in order to capture dual hand manipulation and grasping movements. Observed actions are segmented, categorized, and encoded in the form of multi-layered macro operators structured in a Petri-net for the task representation. The tasks are executed on the humanoid robot ARMAR-II. Using a similar setup as in (Zoellner et al., 2004), (Jäkel, 2013) records grasping movements in the form of wrist trajectories in the task space and fingertip joint angle movements. A probabilistic modeling approach is used to generalize the observations to grasping strategies which are integrated into a probabilistic motion planner. Experiments have been conducted on a dual-arm platform whereas each arm has 7 DoF and is equipped with a 13 DoF hand. In (Ekvall and Kragic, 2004), data glove readings, are stored in a database and categorized according to the observed grasp type. These are mapped to a robot hand using a function learned from data sets which contain human and corresponding robot hand poses. In a subsequent stage, grasp types applied in human grasping actions are recognized and reproduced by the robot using the grasp data. In order to instantiate a DMP for the representation of prehensile joint angle movements, (Amor et al., 2012) used a data

glove to record human grasp examples. The grasp representation is directly mapped onto a four-fingered hand of the humanoid robot and is adapted to the current object to be grasped by contact point warping. In (Kroemer et al., 2010), human grasping movements captured with the Vicon system are used to learn DMPs which encode grasping actions observed in the task space. To adapt the learned DMPs to the current scene, target positions are extracted from a visual object representation based on 3D edges which are acquired using the robot's camera system. The specified grasp primitives are executed on a single arm robot using a Barrett hand. In (Triesch et al., 1999), an approach is presented which allows learning of grasping actions based on observations of human grasp demonstrations acquired by using the onboard sensory system of a robot. To capture human grasping actions, a vision-based hand-fingertip tracking algorithm is applied which enables the recording of the hand and the index-thumb finger movement. The grasping movement is represented in the form of a linear trajectory in the task space and is reproduced using the 7 DoF arm of the robot which is equipped with parallel gripper.

### 7.2. System Overview

According to (Bakker and Kuniyoshi, 1996), imitation learning tasks commonly comprise three different stages: observation, generalization and reproduction. To deal with the problems arising in these stages, various components are incorporated by the proposed framework whose structure is depicted in Figure 7.3.

### 7.3. Mapping

Due to the differences in the kinematic structures of the human subject and a robotic platform e.g. differing joints and limb measurements, in general, a one-to-one mapping does not lead to a goal-directed and natural reproduction of a human movement. Hence, for each human individual, a mapping function has to be specified capable of adapting and optimizing corresponding human motion data to the embodiment of the robot. To minimize the efforts required to motion transfer between different embodiments, a standardized interface by using the Master Motor Map (MMM) has been established which features a high level of flexibility and compatibility.

#### 7.3.1. Master Motor Map

The MMM is introduced in (Azad et al., 2007) and incorporates a reference kinematic model of the human by defining the maximum number of DoF, currently 96, that might be used by any visualization, recognition, or reproduction module. In doing so, a unifying framework is created by defining a data exchange standard which allows the generation of normalized action knowledge structures and facilitates the transfer of motion data between different modules. Besides the kinematic specification of the reference model, the limb segments are enriched with proper body segment properties, such as mass distribution, segment length, and moment of inertia in order to compute gross body dynamics. These anthropomorphic properties are defined and scaled according to linear equations which link these properties to global parameters such as height and weight of the whole body. In (Winter, 2009), based on anthropometric data, a proposition for these equations is inferred. By incorporating these findings, the MMM model can be easily adapted to the actual measurements of the human subject.

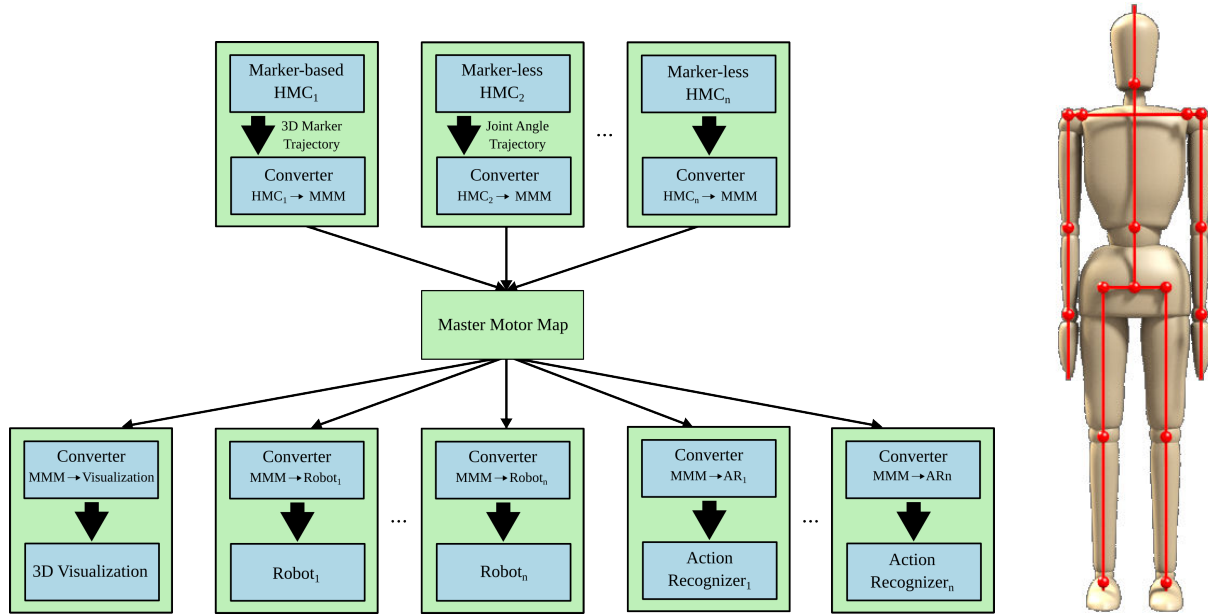


Figure 7.1.: Left: Overview of the Master Motor Map (MMM) framework as introduced in (Azad et al., 2007). Right: Kinematic reference model of the human incorporated in the MMM framework.

Due to differences in orientation of the coordinate systems, in the Euler conventions, and the lengths of the limb segments between the MMM model and the proprietary model of an arbitrary system using the MMM interface, a conversion module has to be implemented for each of the perception systems. This conversion module transforms the module specific data into the MMM file format and vice versa. Similarly, for the reproduction on a robot, a customized conversion module has to be implemented which performs the mapping of MMM data to the embodiment of the robot. Regarding the proposed grasp representation, the correspondence problem has to be solved for the hand approach and the finger movements.

### 7.3.2. Conversion of Fingertip Movements

Since the fingertip movements are captured in the task space, the conversion of captured finger trajectories into an MMM format is accomplished by scaling the trajectories according to the ratio  $r_H = \frac{l_{H_h}}{l_{H_{mmm}}}$  where  $l_{H_h}$  and  $l_{H_{mmm}}$  denote the lengths of the observed grasping hand and the MMM hand. Based on anthropometric data which has been reported in (Buchholz et al., 1992),  $l_{H_{mmm}}$  is defined as a function depending on the global parameters such as full hand height and width. The lengths are measured from the hand's base to the tip of the middle finger. The VSG representation is adapted to the MMM hand measurements by adjusting the equilibrium lengths of the stabilization springs.

To control the fingers of a robotic hand, a mapping based on a geometric inverse kinematics (IK) approach has been implemented which exploits the information that is obtained from the generated grasping strategy. Depending on the grasp type to be executed, whether, volar or non-volar, different information is provided by the VSG representation which can be used to compute the corresponding joint angles. For volar grasps, based on the target fingertip positions, the IK is solved for each finger individually. To do so a plane with a normal vector which is aligned with the joint axis of the PIP joint is spanned between the PIP joint and the tip of the corresponding finger. Based on the finger segment lengths, a trapezoid with equal inner angles is created. In the case of a common palm joint shared between the fingers, the

palm joint is fixed at joint position which is the mean of all IK solutions computed for the fingers. In a second iteration, the IK is solved for reduced kinematic chain leaving out the palm joint. An example of the IK solution for the index and middle finger of the ARMAR-IIIb hand is illustrated in Figure 7.2. Given volar grasp type, additional information in the form of the Virtual Contact Strip is available. Since the finger segments are drawn towards the object surface, a joint angle solution can be immediately inferred.

### 7.3.3. Conversion of Arm Movements

An important condition regarding the mapping arm movements in grasping and manipulation tasks is the preservation of characteristic motion features as well as the goal-directedness. Initially, to normalize marker-based motion data from the specific human embodiment that is observed, the data is converted to MMM using the method introduced in (Gärtner et al., 2010) which is based on virtual markers and an optimization of the joint angle configuration for the reference model. Markerless captured movements are introduced into the MMM by specifying the positions of the MMM joints corresponding to joints which can be tracked using the applied capturing method.

For the reproduction of movements represented as MMM data, in (Do et al., 2008), a procedure is proposed which by means of optimization performs a mapping of human motion in the form of joint angles and reference endeffector positions onto the kinematic structure of a robot. A joint angle configuration is merely considered as an initial kinematic solution which has to be refined in order to bring the endeffector close to the designated target position. Regarding the target positions, once a movement has been generated for the MMM the reference positions are transformed and scaled from an ego-centric coordinate system in the MMM (such as the hip) to the corresponding coordinate system in the robots kinematic chain. Based on these transformed positions and joint angles, the refinement is conducted in two stages: estimation of an initial solution and optimization according to a predefined similarity measure.

### Similarity Measure

One of the most crucial factors in the reproduction of human motion is the measure for rating the similarity between the reproduced and the demonstrated movement. In (Matsui et al., 2005), it is proposed to determine the distance between the postures of the robot and the human by exploiting point correspondences between specified points on both bodies. To infer useful statements concerning the similarity, accurate localization and identification of the limbs are required, which makes the use of physical markers inevitable. In (Zhao et al., 2004), a similarity measure is introduced, which only considers the joint angle relations. However, it disregards structural differences between human and robot like differing limb lengths, which one has to take into account in order to preserve the goal of a movement when mapped on the robot. In this thesis, arm movements are represented in the form of joint angle sequences and task space trajectories. To obtain a natural-looking movement which sustains a minimal loss of goal-directedness, a similarity measure is used that combines both, the joint angle configuration and key point correspondences. Thus, for a joint angle configuration  $\theta \in \mathbb{R}^N$  with  $N$  joints, the similarity measure is defined as follows:

$$S(\theta) = \operatorname{argmin}_{\theta} \left( \left\| \frac{\sum_{i=1}^3 \tilde{\mathbf{x}}(i) - f_r(\theta; i)}{3l_{arm}}(i) \right\| + \left\| \frac{\sum_{n=1}^N \tilde{\theta}(n) - \theta(n)}{N\pi} \right\| \right) \quad (7.1)$$

with  $\theta(n), \tilde{\theta}(n) \in [-\pi, \pi]$  and  $\tilde{\mathbf{x}}(i), f_r(\theta; i) \in [-l_{arm}(i), l_{arm}(i)]$  whereas  $l_{arm} \in \mathbb{R}^3$  describes the maximum extension of the robot arm. The reference joint angle configuration is denoted by



$\tilde{\theta} \in \mathbb{R}^N$ , while  $\tilde{x} \in \mathbb{R}^3$  stands for the desired endeffector position. For the calculation of the current endeffector position, the function  $f_r$  is used to solve the forward kinematics problem for the joint angle configuration  $\theta$ .

### Estimation of an Initial Solution

To obtain a solution leading to a high resemblance to the reference posture, which at the same time meets all the mechanical constraints of the robot, the reference joint angle configuration is optimized regarding the similarity measure as specified in Eq. 7.1. A solution is found by applying a numerical optimization algorithm. To enhance the efficiency of the applied optimization method, an initial guess is to be found which is close to an optimal solution. Such an initial estimation is determined from a preselection of candidate joint angle configurations, which are generated and evaluated by means of the similarity measure. To generate a candidate estimation  $\tilde{\theta}_t$ , the reference joint angle configuration  $\hat{\theta}_t$  computed at time  $t$  is mapped into the robot joint angle space and projected on the bound constraints:

$$\tilde{\theta}_t(n) = \begin{cases} C_{n_{min}} & \text{if } \hat{\theta}_t(n) \leq C_{n_{min}} \\ \hat{\theta}_t(n) & \text{if } C_{n_{min}} \leq \hat{\theta}_t(n) \leq C_{n_{max}} \\ C_{n_{max}} & \text{if } \hat{\theta}_t(n) \geq C_{n_{max}} \end{cases}, \quad (7.2)$$

where  $C_{n_{min}}$  and  $C_{n_{max}}$  denote the lower and upper joint angle bounds of joint  $n$ . If the value of  $\hat{\theta}_t(n)$  exceeds the given bounds, the joint  $n$  is fixed at the closest of the two boundaries. A candidate is obtained by altering each non-fixed joint angle of the mapped configuration by means of a vector  $\delta_t = \hat{\theta}_t - \hat{\theta}_{t-1}$ . Thus,  $\delta_t$  describes the changes between two consecutive frames. As a result, a candidate estimation can be described as:

$$\theta_t^j(n) = \tilde{\theta}_t(n) + \alpha_n \beta_n \quad (7.3)$$

with

$$\alpha_n = \begin{cases} 1 & \text{if } C_{n_{min}} \leq \tilde{\theta}_t^j(n) \leq C_{n_{max}} \\ 0 & \text{else} \end{cases} \quad \beta_n \in \{-\delta_t(n), 0, \delta_t(n)\} \quad (7.4)$$

$$\beta_n \in \{-\delta_t(n), 0, \delta_t(n)\}. \quad (7.5)$$

Given  $N$  joints to control, in the worst case  $M = 3^N$  candidates need to be calculated and evaluated. The best initial estimation satisfies the following equation:

$$\theta_{init} = \underset{\theta}{\operatorname{argmax}} (S(\theta) - \|\hat{\theta}^t - \theta\|). \quad (7.6)$$

### Optimization Problem

For optimization of a reference joint angle configuration with regard to the similarity measure, one can use the LM algorithm briefly described in Section 6.3.3. Due to its numerical stability, the LM method has become a popular method for solving inverse kinematics problems as demonstrated in (Wampler, 1986). For the current problem, where, given the reference joint angle configuration  $\hat{\theta}^t$ , a solution  $\theta^t$  is sought which maximizes Eq. 7.1. To interpret Eq. 7.1 as a function of sum of squares, a residual function  $s(\theta) : \mathbb{R}^N \rightarrow \mathbb{R}$  is inferred from  $S$  as follows:

$$s(\theta) = \frac{\sum_{i=1}^3 \mathbf{x}_H(i) - f_k(\theta; i)}{3l_{arm}(i)} + \frac{\sum_{n=1}^N \hat{\theta}^t(n) - \theta(n)}{N\pi}. \quad (7.7)$$

The corresponding optimization problem can be written in the following form:

$$\begin{aligned} \boldsymbol{\theta}^t &= \operatorname{argmin}_{\boldsymbol{\theta}} (2 - S(\boldsymbol{\theta})) \\ \text{subject to } & C_{n_{\min}} \leq \boldsymbol{\theta}(n) \leq C_{n_{\max}}, \end{aligned} \quad (7.8)$$

which is equivalent to the maximization of Eq. 7.1.  $\boldsymbol{\theta}^t$  is obtained by solving Eq. 7.9 as an unconstrained least squares problem. For a feasible joint angle configuration, the solution is projected onto the bound constraints.

## 7.4. Structuring of Grasp-Related Motor Knowledge

Based on the representations and motion data represented using the MMM framework, the grasp representation is instantiated using the estimation scheme introduced in Section 6.3. To form a grasp primitive, each grasp representation is enriched with additional information relevant for the reproduction on the robot. In addition to the model parameter such as the spring lengths and spring constants, the data structure contains the information about the fingers involved in the data, the chronology of preshape and enclose and whether the encoded grasp is volar or non-volar. However, regarding the acquisition of grasp knowledge, it seems to be obvious to process and learn grasp types which are unknown to the system. To be able to determine whether an observation denotes a grasp type which has already been learned, a model has been trained which allows the classification of finger trajectories.

### 7.4.1. Grasp Type Classification

For the classification of grasp demonstrations, methods are used which have been introduced in (Do et al., 2009) and (Fischer et al., 2011). Based on Support Vector Machines a multi-model classifier has been trained. To reduce the efforts of training, as described in Section 4.1.1 fingertip trajectories are projected to a 2D latent space which has been created using PCA. For each grasp type, a binary classifier is trained by labeling low-dimensional grasp observations featuring this grasp type as a positive training examples whereas remaining grasp examples associated with another grasp type are regarded as negative examples. Based on the resulting training data, a corresponding model is trained which represents the currently observed grasp type. In order to classify a new movement, the fingertip movements are translated in the latent space and classified by fusing the classification results of each model. Models which react on the trajectory input denote grasp primitives where the model parameters can be updated based on the current observation. To update an existing grasp representation, based on the current observation but using the grasp type information encoded in the classified primitive, a grasp representation is instantiated. The parameter set of the primitive is updated by taking the mean of both sets, the stored and the previously estimated one. If no models reacts, a new grasp primitive is created and added to the grasp knowledge base.

## 7.5. Reproduction on the Robot

Given an object located at a position  $\boldsymbol{x}_g$ , the adaptation of a learned grasp primitive is accomplished by, first, parameterizing the DMP representation for the hand approach movement to the object pose and, second, by determining contact positions on the object surface for the VSG representation. The DMP parameterization is based on the assumption that start and goal poses are known and velocities at both ends are zero. For the execution on the robot, the start is denoted by the initial pose of the grasp hand. For the target pose, it is assumed that the object to be grasped is annotated with suitable grasp poses and contact positions

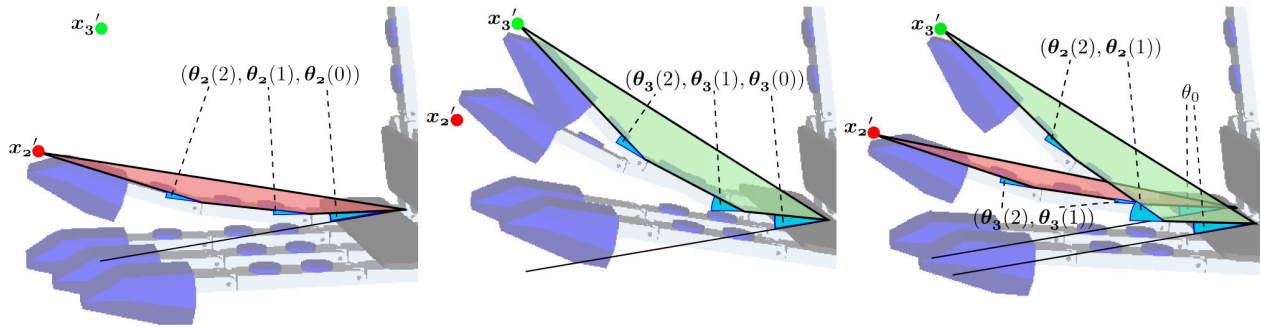


Figure 7.2.: Left: IK solution for the index finger computed based on a geometric approach. Center: Subsequent solution of the IK for the middle finger. Due to the coupled palm joint, the index finger is shifted. Right: Fixation of the palm with the average palm angle value and solution of the IK for shortened kinematic chains (disregarding the palm) for the index and middle finger.

which can be generated by applying appropriate grasp planning algorithms. The contact positions can be used to parameterize the VSG representation. In case, similar objects are to be grasped as featured in the grasp demonstration, the contact positions encoded in the grasp representation can be scaled to meet object-specific requirements. To adapt an instantiated VSG representation to a different object scale, the contact positions originating from human observation are shifted in the direction from the object center to the contact. Hence, the adapted contact point  $v_c$  is determined as follows:

$$v_c = v_m + s(v_c - v_m) \quad (7.9)$$

where  $v_m$  denotes the object center,  $s$  the object scale factor, and  $v_c$  the original contact point for the reference object. Based on the adapted contact information the equilibrium grasp poses for preshape and enclose are modified accordingly, which in return changes the equilibrium lengths of the finger and contact springs. Once both representations have been adapted to object- and task-specific constraints, a grasping movement can be generated by integrating the differential equations incorporated in the grasp primitive.

### 7.5.1. Distance-Based Coupling between Transport and Grip

Due to kinematic inaccuracies and errors in the robot's perception, it cannot be guaranteed that the generated grasping movement leads to actual successful grasp execution. To overcome these deficiencies a visual servoing approach which is introduced in (Vahrenkamp et al., 2008) is applied to make sure that the robot endeffector is aligned with designated grasp pose. However, the subsequent application of the visual servoing method after the hand approach movement is finished requires an additional control mechanism which deals with the prehensile finger movements. Hence a second coupling between hand approach and grip movement is proposed. To accommodate corrective movements such as initiated by a visual servoing algorithm, a coupling is proposed which is based on the distance of the robot endeffector and the object to be grasped. This coupling is mainly accomplished by adding an additional perturbation force to the contact springs which prevent the fingertips from reaching their designated target positions in time. The perturbation term is applied between the virtual finger and the thumb and, thus, translates to external forces which are added to the forces exerted by the

contact springs. Based on a hand-object distance  $d_g = \|\mathbf{x}_H - \mathbf{x}_g\|$ , the external force for a finger  $i$  is described as follows:

$$\mathbf{f}_{i,ext}(t) = \frac{k_{c_i}}{k_{c_1}} d_g \left( \widehat{\mathbf{x}_i - \mathbf{x}_c} \right). \quad (7.10)$$

The ratio between spring constants of the thumb with  $i = 1$  and the remaining fingers is introduced to preserve the prehensile enclosing behavior. To prevent the VSG representation to attain the grasp equilibrium state, Eq. 7.10 is applied before the generated grasping movement is finished.

### 7.6. Summary

In this chapter, a framework is presented consisting of various components which allow the observation, the learning, and the reproduction of different grasp type behaviors featured in human grasp demonstrations. Depending on the used observation method, the framework is capable of learning grasp primitives online or offline. A major focus of this chapter is set on the solution of the correspondence problem between the human and the robot. To be able to generalize the captured motion data from the observed human subject, the MMM framework is incorporated in order to facilitate the transfer of motor knowledge between different embodiments. Hence, the grasp primitives are learned within the MMM framework and used to update the grasp-related motor knowledge structures. Mapping of a grasping action generated from these primitives is accomplished by optimizing the movements using a nonlinear optimization approach. For the execution on the robot, several issues such as the IK solution of the fingers and problems arising from the inaccuracies of the robot are addressed.

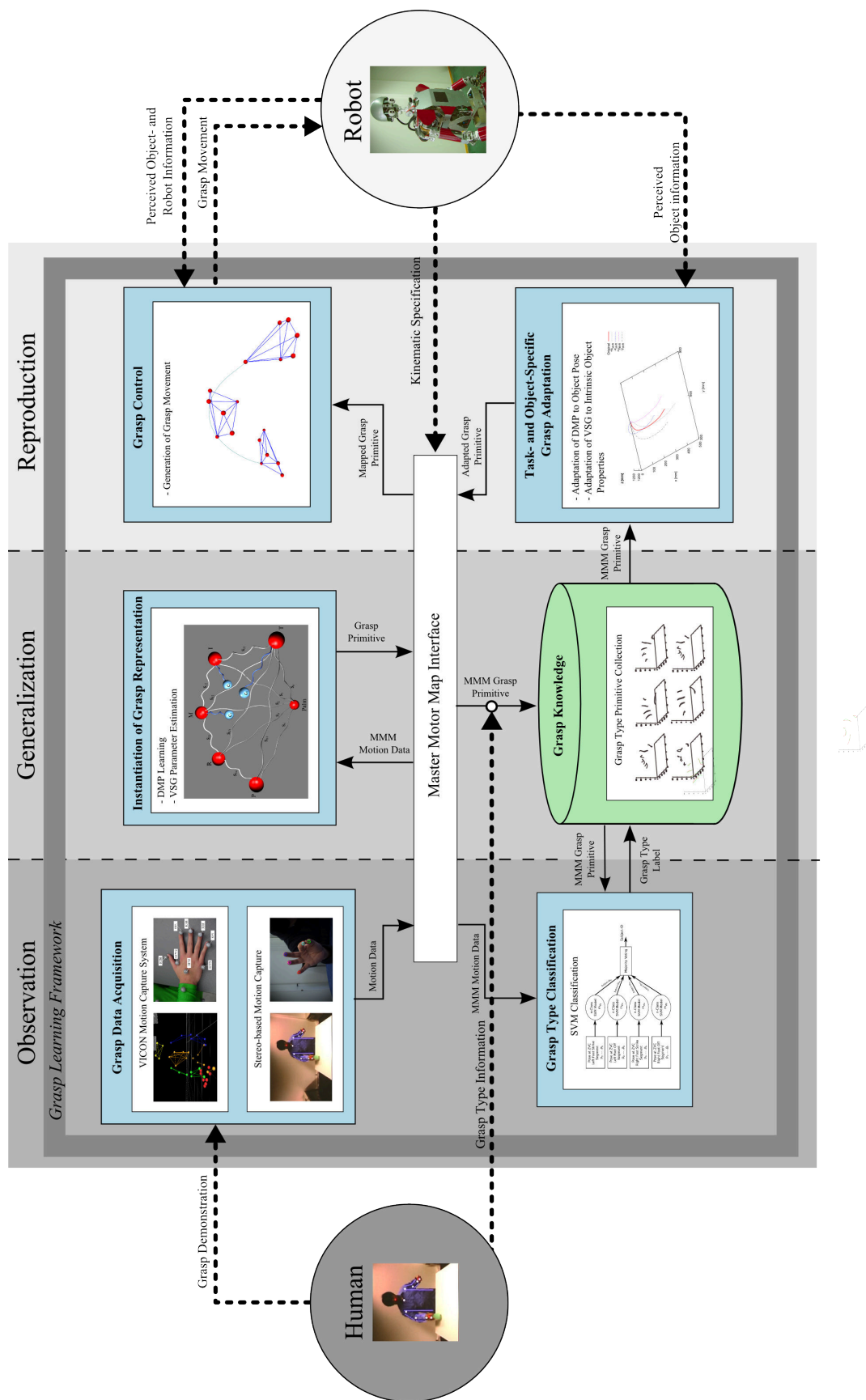


Figure 7.3.: Overview of the proposed framework which employs methods and representations introduced in this work for the implementation of a grasp learning behavior based on human observation, generalization, and the execution on a robotic platform.



## 8. Evaluation

In this chapter, the mechanisms and components integrated in the previously introduced grasp learning framework are individually evaluated in terms of their feasibility and reliability. A major focus of the evaluation is the adaptation and reproduction of grasp primitives which encode human grasp demonstrations using the Virtual Spring Grasp (VSG) representation. To be able to make a statement how accurate human grasps can be represented, most measurements have been conducted in simulation disregarding the potential kinematic errors in the robot platform. The applicability of the proposed approach is shown by experiments carried out on actual robots.

The setup used in these experiments is outlined in Section 8.1. Subsequently, an evaluation of the human observation techniques used in the proposed framework is presented in Section 8.2. In Section 8.4 and Section 8.3.1, the proposed representation is evaluated in terms of aspects regarding accuracy, generalization, and adaptivity. The reproduction and execution of learned grasp primitives is subject to Section 8.5.

### 8.1. Experimental Setup

#### 8.1.1. Experimental Platforms

The humanoid robot ARMAR-IIIb, which serves as the main experimental platform, is a copy of the humanoid robot ARMAR-IIIa (see (Asfour et al., 2006a)). The robot is equipped with two arms whereas each arm features 7 DoF (3 DoF in the shoulder, 1 DoF in the elbow, 1 DoF for the forearm rotation, and 2 DoF in the wrist). At the end of each arm a five-fingered robotic hand is attached. The hand is pneumatically actuated and incorporates 8 DoF (2 DoF for the thumb, middle, and the index finger, 1 coupled DoF for ring and pinkie, and 1 DoF in the palm). For locomotion, the robot has a wheel-based holonomic platform which allows movements in all directions. The main sensory capabilities are integrated in the active head, a subsystem with 7 DoF equipped with two eyes, which have a common tilt and independent can pan units. Each eye is equipped with two digital color cameras, one with a wide-angle lens for peripheral vision and one with a narrow-angle lens for foveal vision. In addition, the active head possesses 6 microphones placed around the head for sound localization and an inertial measurement unit mounted on top of the head. For manipulation purposes, ARMAR-IIIb is equipped with force torque sensors which have been installed between the wrists and the hands. A distinctive feature of ARMAR-IIIb is that the hand system integrates tactile sensor units placed on the palm and on the fingertips which enable the detection of contact.

The second experimental platform consists of a single-arm manipulator equipped with a robotic hand system for fine manipulation. The Motoman-SIA20D arm developed by [Yaskawa Motoman Robotics] is an industrial manipulator. At its end, an anthropomorphic hand system is attached in the form of the Gifu Hand III which is introduced in (Mouri et al., 2002). The hand system incorporates 5 fingers and a total number of 20 DoF. The thumb has 4 joints where each joint is actuated by a separate servomotor. Each other finger has 4 joints which are driven by two servomotors. The root joint which links the finger to the palm allows adduction and abduction movements. The remaining 3 DoF are coupled and enable the flexion of the finger. Since the hand is electrically actuated, an accurate positioning of

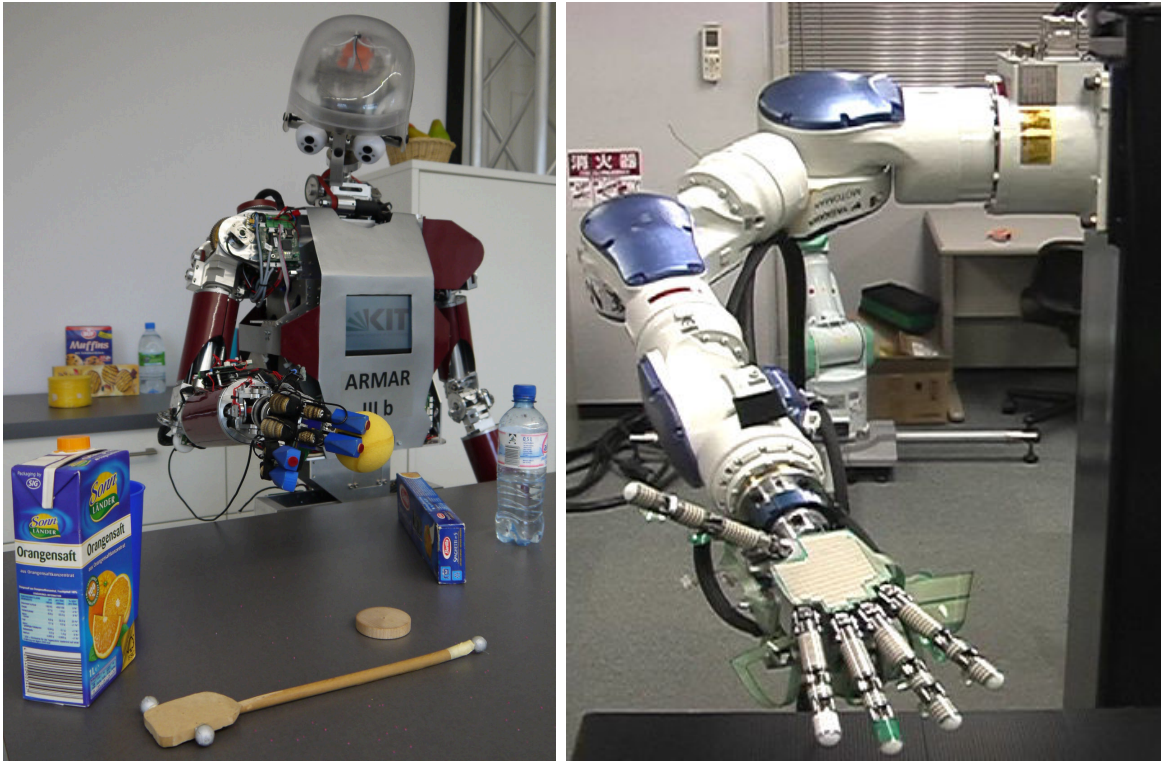


Figure 8.1.: Left: The humanoid robot ARMAR-IIIb with an 8 DoF anthropomorphic hand. Right: The Gifu hand with 20 DoF attached to the Motoman manipulator.

the fingers is attained. However, the system is very sensitive to high forces causing backlashes of the fingers. The platform has served for various grasping experiments as demonstrated in (Ugur et al., 2012).

Both platforms, ARMAR-IIIb and the Motoman/Gifu system, are depicted in Figure 8.1.

### 8.1.2. Object Set

For the learning of grasp primitives, grasp data has been extracted from demonstrations of human grasping actions applied to generic-shaped objects. The shape featured by the objects comprises four different categories: boxes, spheres, cylinders, and discs. The confinement to these objects is motivated by the simplified transferability of grasp knowledge from one object to another. Regarding the learning of non-volar grasps, explicit information about the object's appearance is not required. Object-specific grasp information such as presumed contact positions and the grasp pose are estimated from the demonstration. Therefore, in the following, the intrinsic object properties are reduced to the object diameter which is approximated by the distance between the positions of thumb and the virtual finger in the final grasp configuration. For volar grasps, however, a full object model is required for the computation and the evolution of the Virtual Contact Strip. Each object category contains at least three object instances of different sizes. To distinguish between objects of similar shape but different sizes, an object parameter  $s$  is introduced. Since, the medium-sized objects are considered to be the reference object, the scale for these objects is set  $s = 1$ . The scale parameter for a different object is determined based on the ratio between its diameter and the one of the reference object.



## 8.2. Grasp Data Acquisition

**Marker-based Grasp Observation** Towards a comprehensive grasp repertoire which incorporates common grasping strategies applied in human-centered environments, human grasp demonstrations performed on the generic object set have been recorded using the Vicon system. The selection of observed grasp types is guided by the Cutkosky grasp taxonomy. In the perspective of this thesis, this taxonomy is regarded to be the most universal and compact taxonomy. In the experiments, only single-handed grasping actions have been considered, and, thus, as described in Section 5.2, markers have been attached to the upper body and to the right hand of the human subject. Grasp examples have been collected from the demonstrations of three different human subjects where each subject has been asked to perform a grasp at least ten times. Starting from a common initial pose, the subject grasps an object which has been placed in front, lifts and places the object, and finally returns to the initial pose. The post-processing procedure consists of manual labeling of mislabeled markers and interpolation between discontinuous trajectory segments. To reduce the amount of data that has to be processed, the grasp demonstrations have been captured at a frame rate of 100 Hz which in return resulted in grasp trajectories with approximately 200 – 250 time frames. Throughout this work, approximately 300 grasp demonstrations have been recorded.

**Markerless Grasp Observation** The grasp learning procedure using markerless observation techniques has been adapted to the sensory capabilities of the humanoid robot ARMAR-IIIb. Using both camera pairs of the humanoid’s active head, grasping actions are observed based on image sequences with a resolution of  $R = 640 \times 480$  pixels. The low resolution prohibits the simultaneous tracking of arm and finger movements. Hence, the observation process is performed in two stages where, first, the arm movement of the performed grasp is captured in the peripheral views, and, subsequently, using the foveal views, in a repeated grasp demonstration, the fingertips are tracked. In the first stage, upper body movements are



Figure 8.2.: Images of the tracking results. The upper row depicts simultaneous closing of the fingers, while the lower row shows sequential flexing of the fingers. The fingertips are labeled as follows: Thumb (green), index (light blue), middle (dark blue), ring (pink), and pinkie (red).

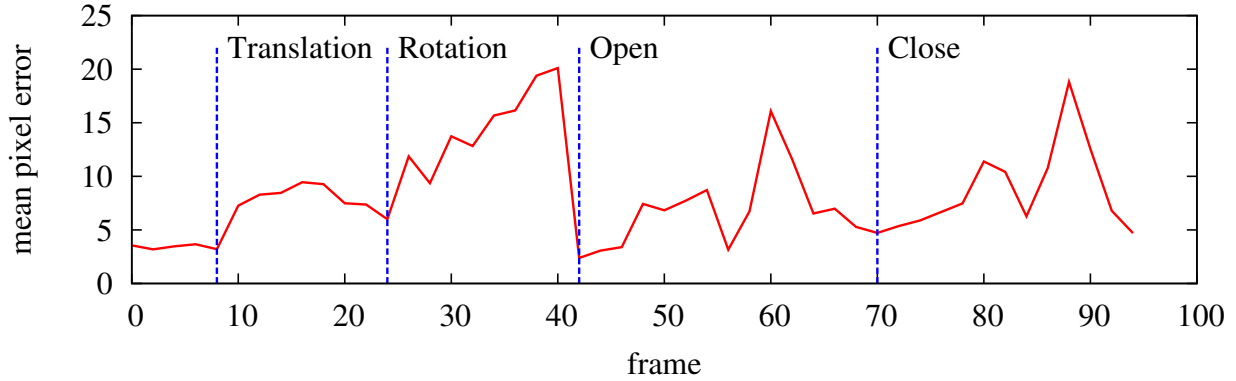


Figure 8.3.: Error plot for a sequence of four hand and finger movements: Translation of the hand, rotation of the hand, close and open movement incorporating flexing the fingers.

recorded using the peripheral cameras of the robot. Based on a known object location  $\mathbf{x}_g$ , the hand approach movement is captured where the grasping hand is identified

In the second stage, during the repeated human grasp demonstration the grasping hand is tracked actively in order to generate the foveated views which enable the tracking of the fingertips. The entire observation process is based on the assumption that the difference between the arm trajectories in the first and the second demonstration is minimal and, thus, can be disregarded. Therefore, the duration of the captured arm movement is temporally stretched to fit the captured fingertip trajectories.

As previously mentioned, markerless captured motion data lacks information based on which the orientation of the grasping hand can be derived. Hence, based on prior knowledge about the presumed initial and final hand poses during the observation stage (which is derived from the orientation of the object to be grasped), an alleged hand orientation trajectory is estimated through interpolation between these poses and added to the motion data. Regarding the selection of grasp types to be learned using the proposed markerless methods, it has been focused on the capturing of precision grasp types since due to the insufficient information, the distinction between volar and non-volar grasp types cannot be made. A one-shot learning behavior allowing the robot to acquire grasp knowledge from a single observation in an online manner is implemented.

A comprehensive evaluation of the upper body tracking framework can be found in (Azad, 2009). In the following, the evaluation of the markerless fingertip tracking approach is described. The fingertip tracker is triggered once the first grasp demonstration is finished. While tracking the grasping hand in the peripheral view, a foveal view is captured based on which edges are extracted at multiple scales  $\sigma_k = 4, 2, 1, 0.5$  and  $\alpha = 0.5$  using the method presented in Section 5.3.5. The fingertip candidates are extracted from the resulting edge map and labeled as described in Section 5.3.5 in order to initialize the tracking procedure.

Once the fingertip tracker has been initialized, equally weighted particles are generated. To increase the robustness of the tracking, the movements of the fingertips are predicted using the trained motion model of the fingertips. In the experiments, the particle filter algorithm performed well with a number of 600 particles allowing to track a single grasping movement without reinitialization. The number of iterations for the subsequent mean shift algorithm depends on the numbers of particles. Thus, the less particles are used the more mean shift iterations are needed. For 600 particles, the mean shift converged after approximately 3 iterations. The accuracy of the fingertip tracking method has been evaluated based on deviations in terms of image coordinates. The tracking framework has been evaluated for different movements such as translation and rotation of the entire hand and common opening and finger

closing movement. The fingertip positions have been estimated with a mean deviation of  $\approx 7$  pixels for translational movements. For a rotational, opening and closing movement, the error increases up to 20 pixels. These measurements are depicted in Figure 8.3. The tracking procedure is considered to be failed once a finger is lost, which is the case when a finger is occluded or its movements are too fast.

In case of failure, a reinitialization is performed consisting of a search for maximum bins in the vicinity of the last known estimation and arrangement of the fingers according to the positions in polar space. Since this algorithm operates on monocular images, for each view, a tracking instance is created whereas the 3D positions of the fingers are calculated by exploiting the epipolar geometry. A limitation of the proposed tracking framework is that, for the initialization, the palm has to be visible and facing the camera. Furthermore, due to increased motion blur by using the active head, the human subject is asked to perform prehensile movements at a lower velocity (approximately half of the natural speed). As it can be observed in the sample images depicted in Figure 8.2, the proposed algorithm accurately tracks the fingertips when the fingers are flexed. A problem which occurs during the closing movement is that occasionally the finger knuckles are detected instead of the fingertips. However, using the onboard sensors of a humanoid robot, the framework allows the tracking of the fingertips with a frame rate of 25 Hz on a 2.40 GHz quad-core CPU.

### 8.3. Grasp Modeling

As stated in Chapter 4, the proposed grasp representation consists of two models of different granularity. The hand approach movement is represented in the form of a Dynamic Movement Primitives (DMP) which is learned from the captured motion data using the Receptive Field Weighted Regression method described in Section 6.5. Approximately 10–20 basis functions are needed for the encoding of the hand approach movement. A comprehensive analysis on the performance of the Receptive Field Weighted Regression method is presented in (Schaal and Atkeson, 1998). With regard to the representation of prehensile representation movements, the focus is set on the question how well the proposed VSG approach performs compared to previously proposed continuous task-space representations. For this purpose, various grasp primitives are derived from human grasp demonstrations and evaluated in order to address the issue of how accurate grasp behaviors can be encoded and how well the resulting grasp primitives can be adapted to different object properties. In this context, it has been mainly focused on a comparison with other approaches based on dynamical systems such as the DMP-based approach presented in (Kroemer et al., 2010).

#### 8.3.1. Grasp Parameter Estimation

Here, it is focused on the evaluation of the optimization scheme which has been described in Section 6.3 for the instantiation of the VSG representation. For this purpose, experiments have been conducted based on simulated as well as on the captured fingertip motion data. To generate simulated grasping movements, the VSG representation has been parameterized with an arbitrary set of spring constants and spring lengths which originate from the MMM hand measurements. Modulation at a frequency of 40 Hz resulted in fingertip trajectories with a duration of 2 s which in return resulted in simulated grasp data consisting of 80 frames. The preshaping behavior is simulated using a Gaussian-based perturbation term that is centered around  $T_{pre} = 0.66 \cdot T_{end}$  and creates a repelling force which emanates from the center of the system and is applied on each fingertip.

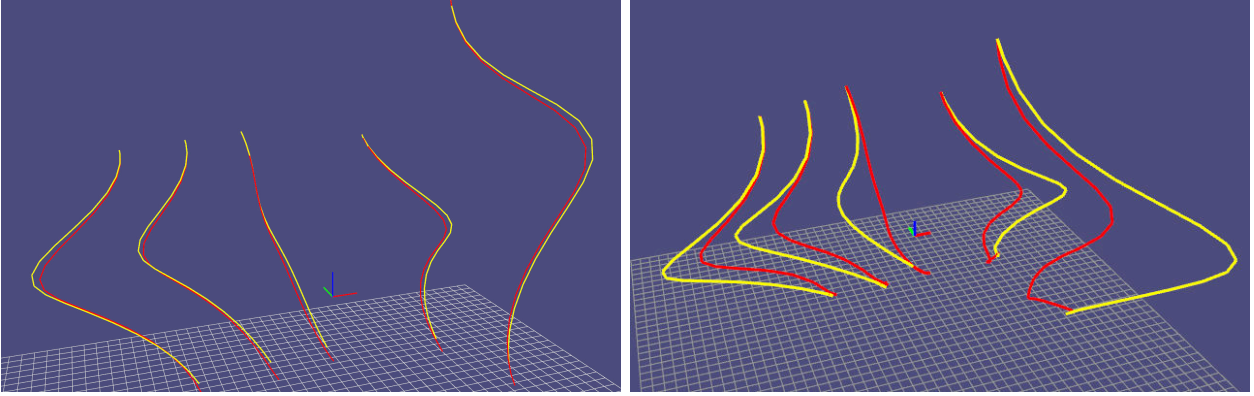


Figure 8.4.: Left: Trajectories of simulated (red) and reproduced (yellow) finger movements using parameters estimated by the proposed optimization scheme. Right: Simulated and reproduced trajectories using parameters estimated by a nonlinear gradient-based optimization algorithm initiated by the proximity sampling step.

In order to employ the parameter estimation scheme, the simulated grasp sequence has been subdivided into  $N_S$  subsets. A subset size with 20 frames has been specified. Based on each subset, observations are formed as described in Section 6.2.1 which serve as input to the estimation procedure. To diminish the risk of falling into local minima,  $\approx 100$  possible solutions which are obtained by sampling the vicinity of previous estimate (from the preceding subset) are evaluated. The best solution is used as an initial estimate for the Total Least Squares solver (TLS). By iteratively solving the optimization problem for each subset,  $N_S$  local parameter estimates are calculated. Infeasible local solutions are discarded while feasible estimates are fused to a global solution. Solutions which are close to a null solution or which deviate 20 % of the mean solution are denoted as infeasible. When performing the optimization on simulated data, it has been observed that with smaller subsets estimation results become more accurate. As depicted in Figure 8.5, the spring constant optimization for each subset converged after 13 iterations. The fused global solution deviates  $\approx 1\%$  from the reference solution. Due to accurate local estimates, only a single iteration of the entire optimization scheme is needed. In Figure 8.4, it can be seen that the reproduced trajectories using the spring constants estimated by the proposed optimization scheme feature a higher similarity to the simulated movement compared the parameters estimated by a non-iterative scheme based on gradient-based optimization algorithm which only considers the spring constants as subject to be optimized.

### Parameter Estimation from Motion Capture Data

Regarding the learning of grasp primitives, it is more interesting to investigate the performance of the proposed parameter estimation scheme on real motion data. Due to the increased level of noise, small-sized subsets lead to heterogeneous solutions. Therefore, a unique global solution is difficult to extract. In general, the more frames which are considered within a subset, the larger the inconsistencies within data become, yielding spring constant estimates around zero. The subset size depends on the data which is to be processed and, thus, a reliable statement on the adequate number of subsets cannot be made. The subset size has been chosen in order to maximize the number of feasible local solutions and, therefore, have been determined experimentally.

The parameter estimation from marker-based motion capture data, has been performed with a subset size of 20 – 25 frames. The initial sampling procedure has been conducted with a number of 1,000 samples whereas in subsequent iterations, the number of samples has

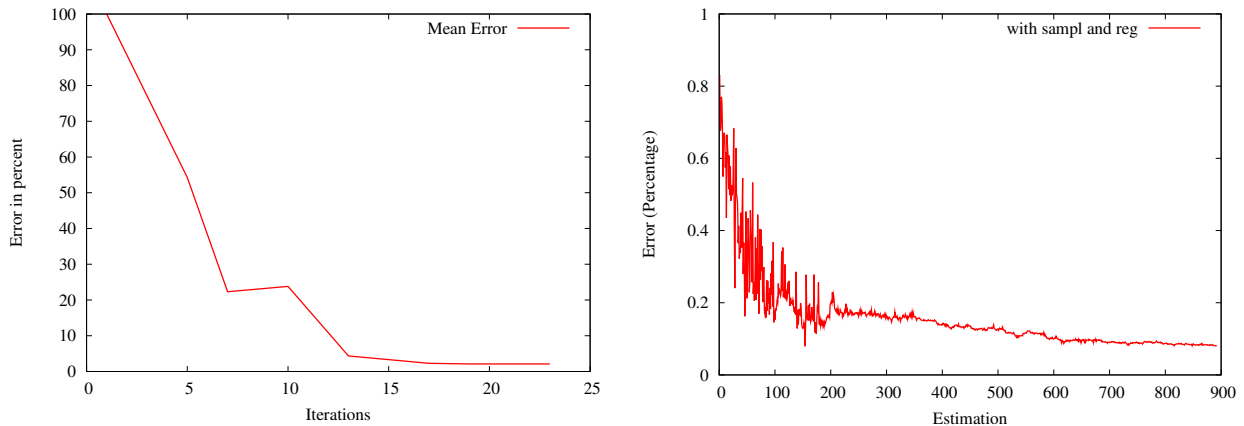


Figure 8.5.: Left: Convergence behavior of the entire estimation scheme on marker-based captured motion data. Right: Convergence behavior of the TLS solver for estimation of the spring constants estimates from marker-based and markerless captured motion data.

been reduced to 100. The computational time needed for the sampling procedure is less than 100 ms and decreases to and, thus, has a minimal effect on the overall runtime. The inner loop which computes the solution of the TLS problem converged after approximately 5,084 iterations. The maximum number of iterations has been restricted to 10,000. The entire estimation scheme required 3 iterations to converge to a global solution. Since each grasping movement has been subdivided in 10-15 subsets the total computation time amounts to 720 s.

Regarding the estimation of spring constants based on motion data extracted from stereo camera sequences, the parameter estimation proved to be difficult due to low frequency at which a grasping movement is captured (limited to 20 Hz according the tracking performance) and the inaccuracies in the position estimations originating from the visual tracking. These deficiencies causing more uncorrelated noise within the data lead to increased number of null solutions for larger subset sizes. Hence, to overcome these problems, the number of samples used to refine the initial parameter estimate has been raised to 5,000 whereas for subsequent iterations 1,000 samples are considered adequate. Furthermore, the subset size has been reduced to 8 – 10 frames. However, due to the smaller subsets the optimization featured a faster convergence. Hence, 511 iterations of the TLS solver is required whereas the entire estimation

Method	Markerless			Marker-based		
	nIterations	Runtime	nSolutions	nIterations	Runtime	nSolutions
MMA	1000	190228	27	10000	189135	90
PTNEWTON	679	187278	45	7978	126388	92
LBFGS	516	107026	31	8472	138015	46
SLSQP	3	302	0	22	3573	7
VAR	582	113127	63	5553	92443	94
Proposed	511	4948	59	5084	16323	77

Table 8.1.: Comparison between the proposed parameter estimation scheme and gradient-based optimization algorithm implemented in the NLOpt library. Results are listed for the estimation of 100 subsets originating from the processing of 8 grasp trajectories.

scheme does not feature any significant change after 3 iterations. For grasp trajectories with the size of 80 – 100, the computation time amounts up to 150 s. The parameter estimates have been calculated in an offline manner using a 2.40 GHz quad core CPU. By comparing the parameter estimation from simulated and real data, a large discrepancy between both results becomes apparent where its reason mainly lies in the modeling error due to the simplification of the hand structure.

The estimation scheme has been evaluated against various multivariate gradient-based regression algorithms implemented in the NLOpt library which is available at [Johnson]. As listed in Table 8.1, the following algorithms have been employed: Method of Moving Asymptotes (MMA), Preconditioned truncated Newton (PTNEWTON), Low-storage Broyden-Fletcher-Goldfarb-Shanno (LBFGS), Sequential Least-Squares Quadratic Programming (SLSQP), and the Shifted Limited-Memory Variable-Metric algorithm (VAR). For the given problem, the SLSQP algorithm which is based on a quasi-Newton method showed the worst performance by fastly converging towards a null solution. Regarding the number of feasible solutions obtained by the applied optimization algorithms the proposed scheme is outperformed by the VAR and the MMA. Regarding marker-based captured motion data, more than 90% of the solutions computed by these algorithms have met the imposed feasibility criteria, while the application of proposed scheme led to 77% of feasible solutions. On markerless motion data, the performance of the VAR and MMA algorithms drastically decreased leading to only 27 respectively 45% of feasible solutions. In comparison to these results, using the proposed estimation method 59% of the calculated spring constant estimates calculated from markerless observations have been considered feasible. Regarding the runtime the scheme performs exceptionally well compared to the other approaches. The main reason for this performance lies in the separation of the estimation problem, since the computational costs grow exponentially with  $O(MN^3)$  with  $N$  denoting the dimensionality of the problem and  $M$  being the subset size.

### Spring Constant Estimates

The final spring constant estimates extracted from grasp demonstrations which feature various grasp types defined in the Cutkosky taxonomy are depicted in Table 8.2. Referring to the finger spring constants  $\mathbf{k}_f$ , one can observe that the grasp behavior is mainly influenced by the first four components which denote virtual springs linking the thumb to the remaining fingers. In particular, the third and fourth component become increasingly important for grasp types where the opposition of the virtual finger and thumb is crucial. With regard to contact and the stabilization springs, it can be observed that the stiffness parameters related to the thumb are scaled larger than those of the virtual finger. This is due to the prominent role of thumb which originates from the guiding function and the larger distances that this finger has to cover. The spring constant solutions between different grasp types such as Spherical Power, Spherical precision feature a high similarity. On the one hand, it means that the information indicating whether a volar or a non-volar grasp is to be learned cannot be directly extracted from mere motion data. On the other hand, this result leads to the finding that similar movements lead to similar solutions which supports the assumption that the proposed estimation scheme parameterizes the VSG representation depending on the featured grasp type rather than the explicit demonstration. To underline this finding, in Table 8.3, solutions are listed for different trajectories which represent the same grasp type.

Due to the missing of reference values, the quality of the parameter estimates can only be assessed by comparing the reproduced and the demonstrated movements. For the trajectory comparison, the instantiated VSG representation has been used to reproduce prehensile move-

ments under the same conditions as specified in the demonstration. The resulting trajectories are depicted in Figure 8.7 showing validity of the grasp representation approach.

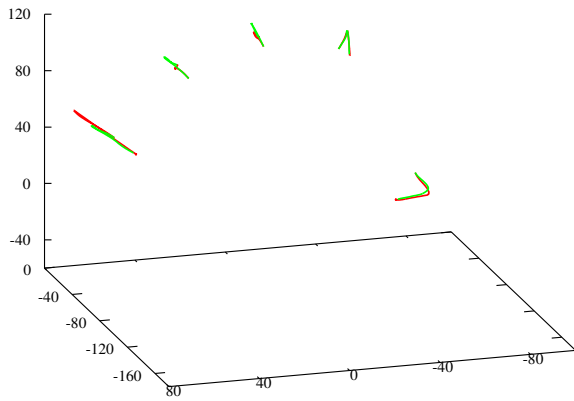
However, the similar shapes of the reproduced and the observed trajectories indicate that the main characteristics of each grasp type could be retained, and, most importantly, grasps could be reproduced in a goal-directed manner since the fingertips involved in the grasping tasks are aligned with their contact positions while non-involved fingers remain at a designated distance to the object. The deviation between observation and reproduction becomes maximal when the fingers attain the maximum grip aperture. Grasp primitives learned from marker-based captured motion data which involve all fingertips could be reproduced with an maximum mean deviation of 5 mm whereas for three and two finger grasps this error increases to 7 respectively 10 mm. For motion data captured with the stereo camera system, the error adds up to 23 mm. This phenomenon can be traced back to the small number of springs. Fingertips not participating in a grasping action are only anchored to the hand structure yielding a wider range of motion, while, for a grasping finger, a smoothing effect is attained which prevents a detailed representation of a grasping movement. In general, it could be observed that by introducing more mass spring damper systems into the dynamical system, the more accurate a grasp behavior can be represented and reproduced.

To evaluate the proposed representation for prehensile fingertip movements in terms of accuracy the VSG representation has been compared against a probabilistic and a dynamical systems approach. With regard to probabilistic representations, grasp primitives are derived using the method which has been introduced in (Romero et al., 2010). From multiple demonstrations of the same grasp behavior, using the Principal Component Analysis a latent space  $F' \in \mathbb{R}^3$  (two-dimensional feature space and one temporal dimension) is inferred which allows the low-dimensional description of a high-dimensional fingertip configuration defined in  $F \in \mathbb{R}^{15}$ . Based on this space 50 Gaussian Mixture Models have been trained in order to obtain a generalized representation of the demonstrated trajectories. Through regression and backprojection the encoded grasp behavior can be synthesized.

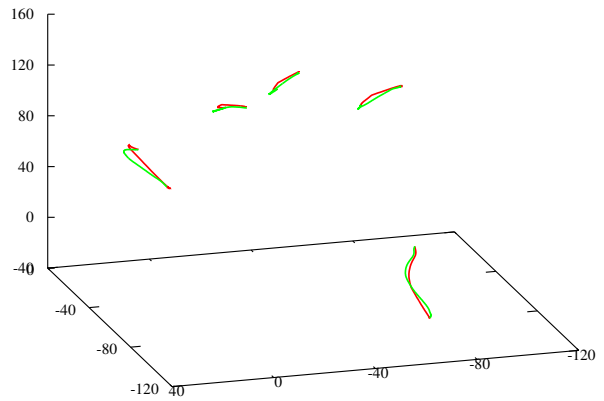
To compare the VSG representation to another dynamical systems approach, based on the method introduced in (Kroemer et al., 2010) DMP representations for fingertip movements have been created. This DMP representation incorporates a transformation system for each

Grasp Type	$k_{c_0}$	$k_{s_0}$	$k_{c_{vf}}$	$k_{s_{vf}}$	$k_f$
Spherical Power	41.00	21.50	31.40	3.44	(9.61, 15.99, 13.59, 29.66, 1.00, 1.00, 1.00)
Circular Power	75.22	1.00	31.70	2.54	(7.10 10.98 9.51 25.02 6.19 5.74 21.81)
Heavy Power Wrap	51.41	13.44	32.69	11.68	(1.00,1.00,1.00,20.00,1.00,5.79,1.00)
Medium Precision Wrap	111.22	38.69	88.54	6.00	(1.00,1.00,1.00,44.48,1.00,1.00,1.00)
Adducted Wrap	79.84	1.00	27.07	18.04	(37.29, 23.47, 11.23, 32.75, 1.00, 30.50, 8.89)
Prismatic Wrap	34.95	18.60	39.37	1.77	(16.74, 18.27, 1.00, 6.41, 1.00, 1.00, 14.70)
Spherical Precision	40.00	5.96	31.03	5.03	(11.42, 20.75, 24.92, 36.47, 1.30, 1.00, 1.39)
Circular Disk	41.00	3.59	31.40	3.44	(9.61, 15.99, 13.59, 29.67, 1.00, 1.00, 1.00)
Thumb-4-finger	45.97	6.28	37.22	1.50	(1.00, 2.94, 16.12, 16.97, 1.09, 12.18, 1.00)
Thumb-3-finger	38.80	7.20	35.43	1.70	(11.14, 21.41, 30.33, 11.72, 3.69, 7.23,1.00)
Tripod	17.21	28.04	18.93	2.08	(1.00, 1.00, 11.08, 16.23, 1.00, 1.00, 1.00 )
Pinch	222.72	14.74	138.04	7.01	(57.26, 122.52, 107.08, 75.41, 6.43, 1.00, 1.00 )

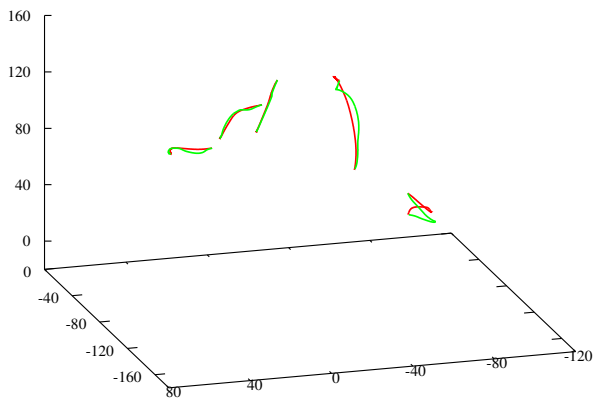
Table 8.2.: Spring constant estimates representing various grasp types.



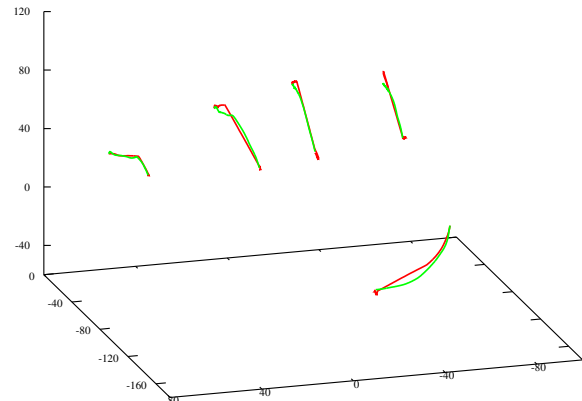
(a) Spherical power grasp



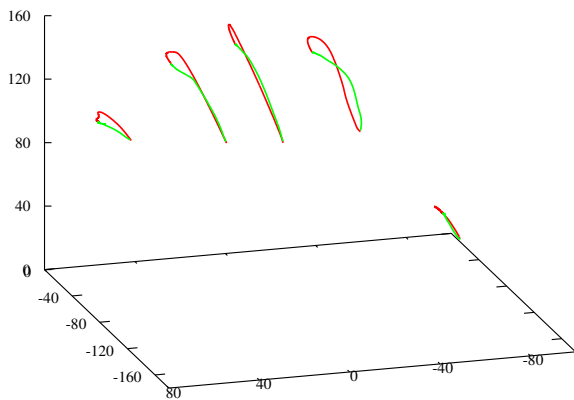
(b) Circular disc power grasp



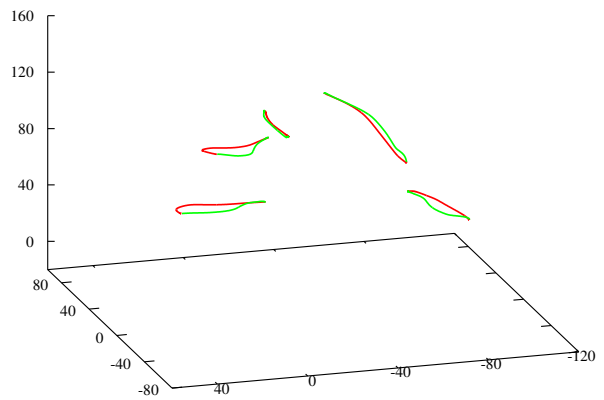
(c) Heavy wrap grasp



(d) Medium wrap grasp



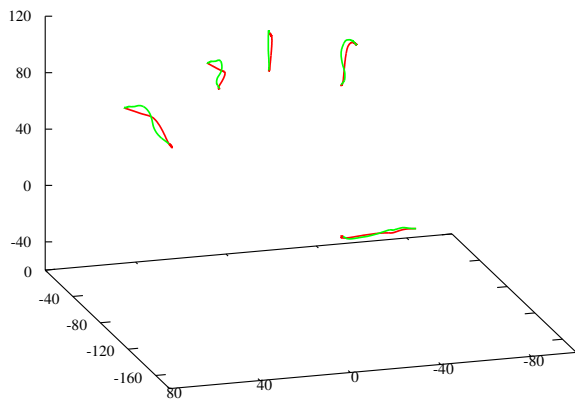
(e) Adducted thumb grasp



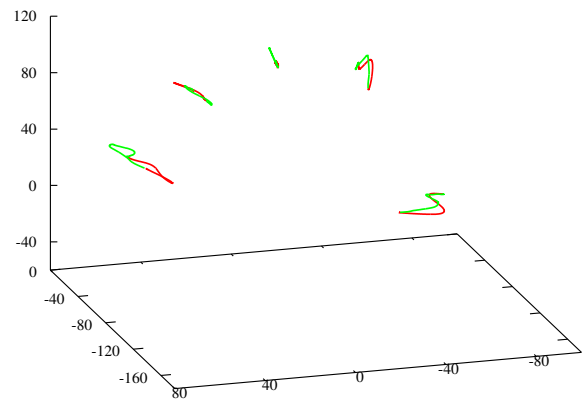
(f) Prismatic light tool grasp

Figure 8.6.: Trajectories of original (green) and reproduced (red) prehensile fingertip movements generated from grasp primitives which encode volar grasp types using the VSG representation. In each plot, the most right trajectory represents the thumb movement whereas the most left depicts the movement of the pinkie. The measurements are given in mm.

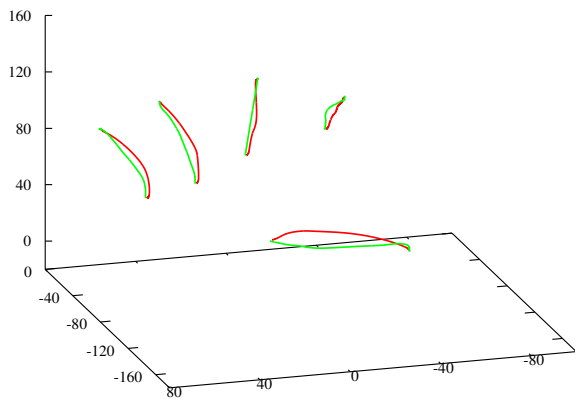




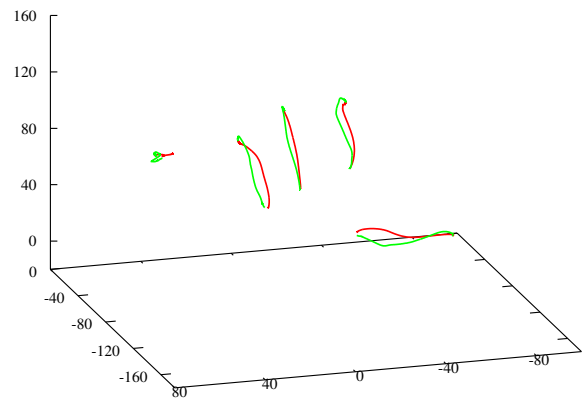
(a) Spherical precision grasp



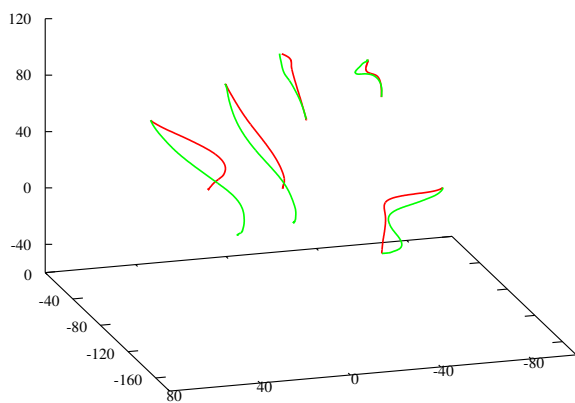
(b) Circular disc precision grasp



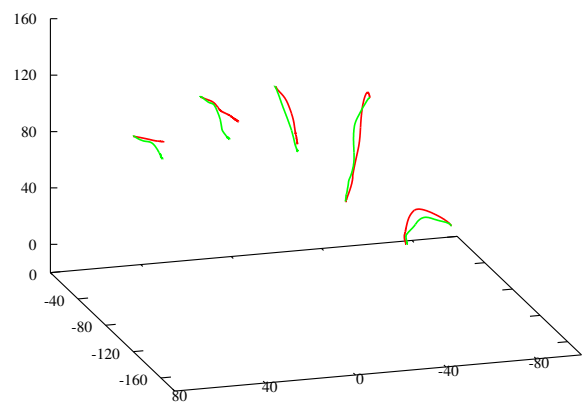
(c) Thumb-4-Finger grasp



(d) Thumb-3-Finger grasp



(e) Tripod grasp



(f) Pinch grasp

Figure 8.7.: Trajectories of original (green) and reproduced (red) prehensile fingertip movements generated from grasp primitives which encode non-volar grasp types using the VSG representation. In each plot, the most right trajectory represents the thumb movement whereas the most left depicts the movement of the pinkie. The measurements are given in mm.

Grasp Type	$k_{c_0}$	$k_{s_0}$	$k_{c_{vf}}$	$k_{s_{vf}}$	$k_f$
Spherical Power 1	70.90	13.11	43.97	1.41	(8.46, 12.50, 13.50, 20.65, 5.89, 1.53, 1.00)
Spherical Power 2	41.00	21.50	31.40	3.44	(9.61, 15.99, 13.59, 29.66, 1.00, 1.00, 1.00)
Spherical Power 3	38.74	23.03	33.76	1.0	(11.67, 12.19, 14.07, 31.78, 1.56, 1.00, 1.00)
Spherical Power 4	42.85	24.51	34.22	17.05	(10.49, 17.48, 13.31, 30.14, 2.27, 1.56, 1.00)
Tripod 1	12.39	23.86	14.56	2.24	(1.00, 1.00, 10.59, 14.27, 1.00, 1.00, 8.23 )
Tripod 2	13.75	24.38	18.75	8.49	(1.00, 1.00, 15.64, 10.37, 1.00, 1.00, 6.55 )
Tripod 3	16.21	22.67	17.09	2.17	(1.00, 1.00, 12.20, 16.49, 1.00, 1.00, 1.00 )
Tripod 4	17.21	28.04	18.93	2.08	(1.00, 1.00, 11.08, 16.23, 1.00, 1.00, 1.00 )

Table 8.3.: Spring constant estimates for similar grasp examples.

dimension, hence, only focusing on positional information, 15 transformation system have been employed where each incorporates approximately 10 basis functions. Thus, 150 parameters have been determined to represent movements in a continuous fashion. Similar to the hand approach movement, the nonlinear perturbation term incorporated in the DMP representation is used to shape an attractor landscape where the attractor is not necessarily a contact point. As depicted in Figure 8.12, based on these representation, various primitives for the encoding of spherical precision are generated and synthesized. The movement reproduced from the GMM-based representation featured a mean deviation of 9 compared to the demonstrated grasping movement. However, in contrast to the dynamical systems approaches, less smooth fingertip trajectories have been generated which is indicated by the jittery error trajectory. Due to the lack of dynamic information such as velocity and acceleration in the GMMs, the generated trajectories contains points which are less correlated than in trajectories originating from the evolution of a dynamical system. In addition, the fact that the designated contact positions are not exactly aligned with the fingertips implies that GMM-based primitives do not provide the inherent capability of encoding the relationship between the endeffector and the object. As opposed to this, attractors in dynamical systems allow the incorporation grasp relevant information such as contact points in order to attain a goal-directed representation and, thus, allow the creation of flexible and adaptive grasp primitives which are necessary for efficient grasp synthesis in dynamic environments.

Clearly, the DMP representation allows a more accurate representation of the demonstrated finger movements. For all grasp types, independent of the fingers involved in the grasp, a deviation of maximum 5 mm has been attained. This error decreases with larger number of basis functions that is used to describe the nonlinear perturbation term. However, in contrast to the proposed VSG representation, DMPs for grasping do not encode the relations between the hand and object properties. As shown in Section 8.4.2, this feature becomes increasingly important when encoded grasp primitives are to be adapted to different situations.

#### 8.4. Grasp Adaptation

Since the proposed grasp representation consists of two models for the hand approach and the prehensile finger movements, separate experiments have been conducted focusing on the evaluation of the representation for the hand approach movement as well as the finger movements. With regard to the hand approach movement, the DMP representation has been evaluated concerning the applicability to represent various discrete arm movements in reaching and

manipulation actions in a goal-directed manner allowing the adaptation to different extrinsic object properties (such as position and orientation). The evaluation of the VSG representation is focused on how the incorporated dynamical systems behave on changes regarding the intrinsic properties of an object (such as size and shape).

### 8.4.1. Adaptation of Arm Movements

To study the representation of discrete arm movements, a simple pick and place scenario has been implemented as an imitation learning task. Based on the experimental setup as described in Section 8.1 human arm movements have been recorded showing the execution of a pick a place task. For this scenario, the segmentation led to motion segments representing three different actions: hand approach, object placing, and hand retreat. For each action, a class of DMP strategies is learned. Due to the simplicity of the movement, a total number of 10 – 20 basis functions is regarded to be sufficient for the approximation of the nonlinear perturbation term. Concerning the approach and retreat movement, each class contains two different DMP strategies assuming that the object position may vary along the vertical axis (either right or left relatively to the endeffector). For the placing action, four DMPs have been trained in order to accommodate placing of objects from back to the front, from left to right and vice versa. As depicted in Figure 8.8, the pick and place task has been reproduced under different conditions by shifting the target positions for each DMP.

A more complex imitation learning task based on the pick and place-scenario has been implemented in order to enable a robot to learn the shell game. In addition to existing DMP set, sliding movements were demonstrated to the robot. For this purpose, the human user has been asked to slide an object in eight different ways. Four distinct movements have been identified which are characterized whether the object has been moved from left to right, away or towards the robot. Adding the four sliding DMPs to the library, a set of movement primitives is obtained which cover the motion needed for performing the shell game. Each DMP is labeled with the semantic movement description (such as {*”approach”*, *”transport”*, *”retreat”*}) and information about the coarse movement direction (*”left\_to\_right”* or *”front\_to\_back”*).

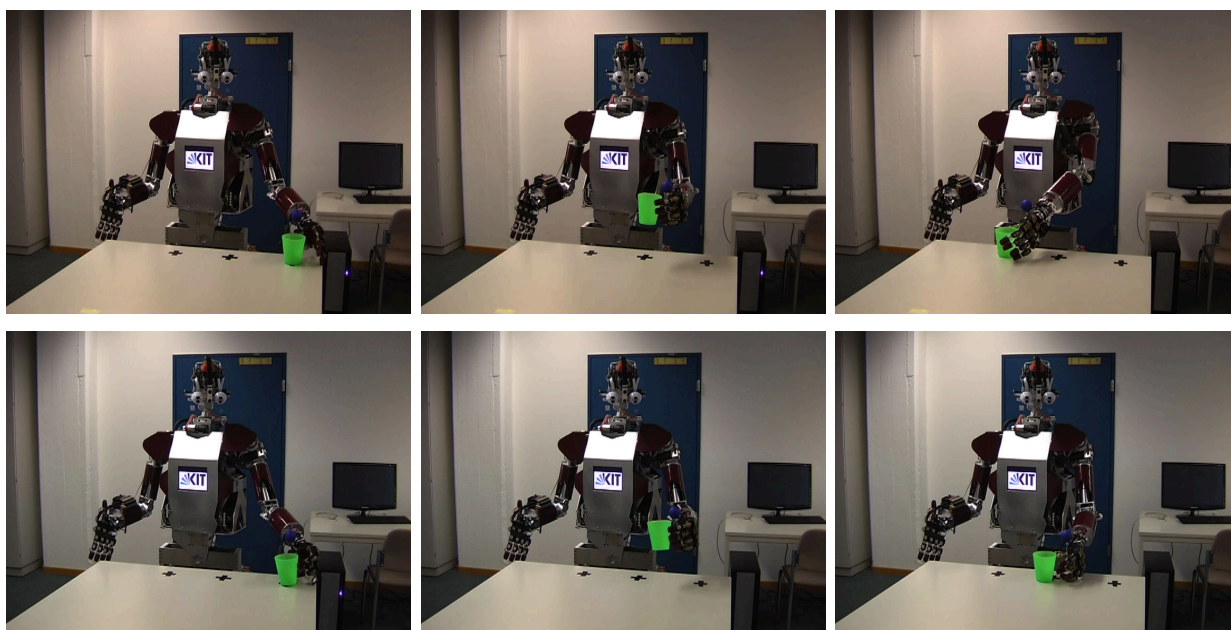


Figure 8.8.: Adaptation and reproduction of DMPs representing a pick and place task learned from human observation on the humanoid robot ARMAR-IIIb.

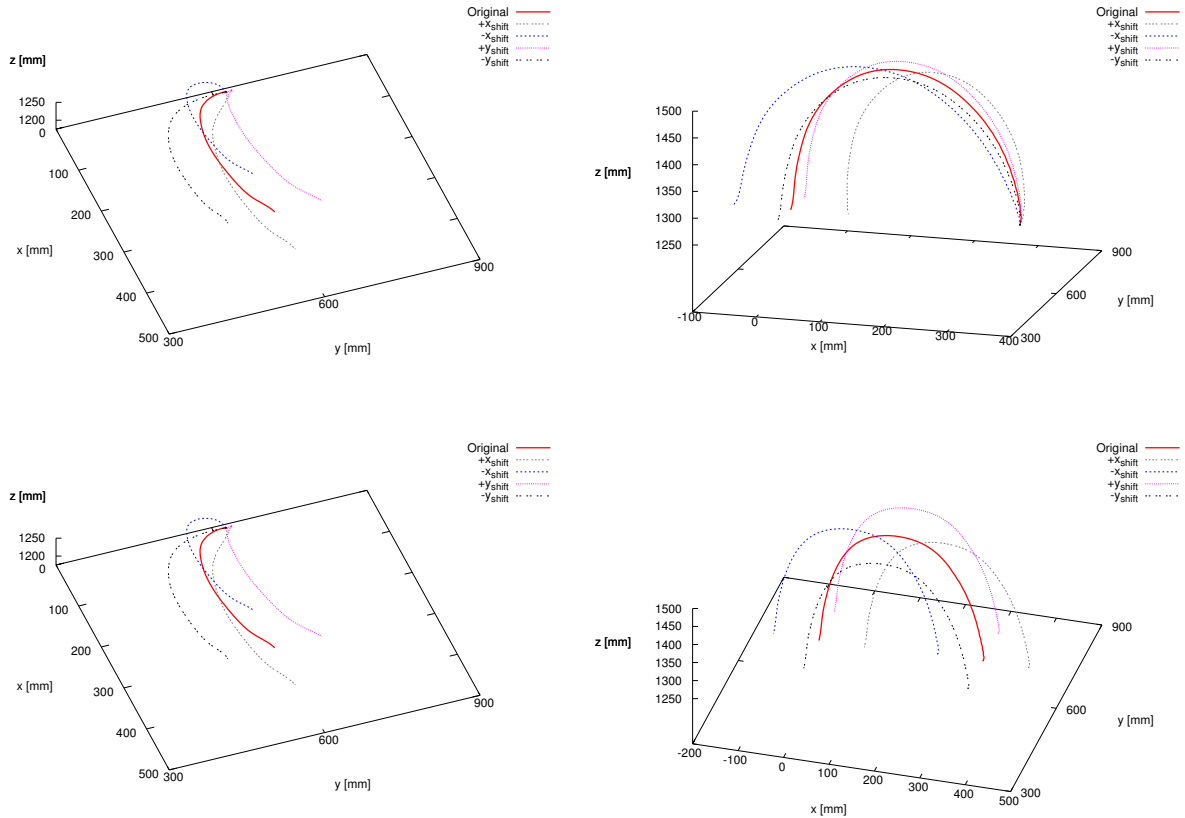


Figure 8.9.: Reproduced DMP movements for different actions: sliding, placing, and the adaptation to different target, and start position.

Assuming zero velocity and acceleration at start and goal, for the reproduction of a complex task, a sequence of chained DMP labels and target poses is passed to the robot. In the case of the shell game scenario, the results of the reproduction are depicted in Figure 8.10.

The movement primitives successfully converged towards the target conditions. However, a problem that has been encountered is that the movement slows down considerably in the vicinity of the target pose which is due to shallow decay of the velocity profile caused by the constant damping factor. Nonetheless, regarding the encoding and reproduction of discrete arm movements where goal-directedness primarily matters this issue can be disregarded.

### 8.4.2. Adaptation of Finger Movements

In the following, the ability of the VSG representation to adapt to different object-specific constraints is investigated. The adaptation to different predetermined contact positions is guaranteed due to contact springs with zero length which enforce alignment of the mass bodies representing fingertip and contact point. Hence, in this evaluation, it is rather focused on the adaptivity of the VSG representation to different object scales. In this context, an informative cue is given by the evolution of the grip aperture movement. To adapt an instantiated VSG representation to a different object scale  $s$ , the contact positions originating from human observation are shifted in the direction from the object center to the contact. For volar as well as non-volar grasps, the equilibrium spring lengths of the finger springs and the contact springs are to be scaled by the factor  $s$ . Similar to the problem of the DMP formulation, the convergence towards the spring equilibrium in the preshaping phase is slowed down prohibiting to reach the full grip aperture even at slower reproduction velocity. To minimize this effect,

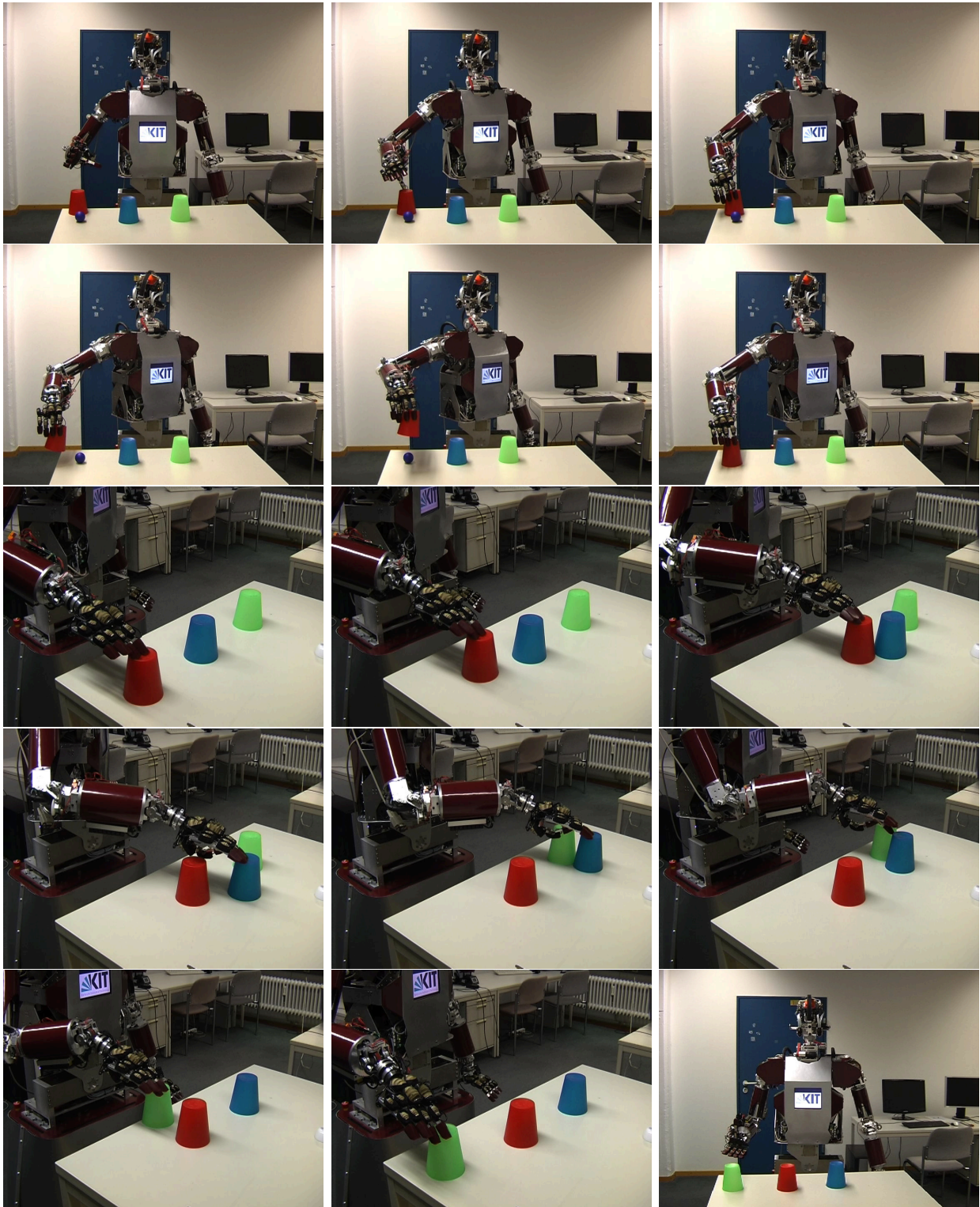


Figure 8.10.: Reproduction of the shell game scenario encoded as a DMP and learned from human observation. The movement has been adapted to different target positions shifted in front of the robot.

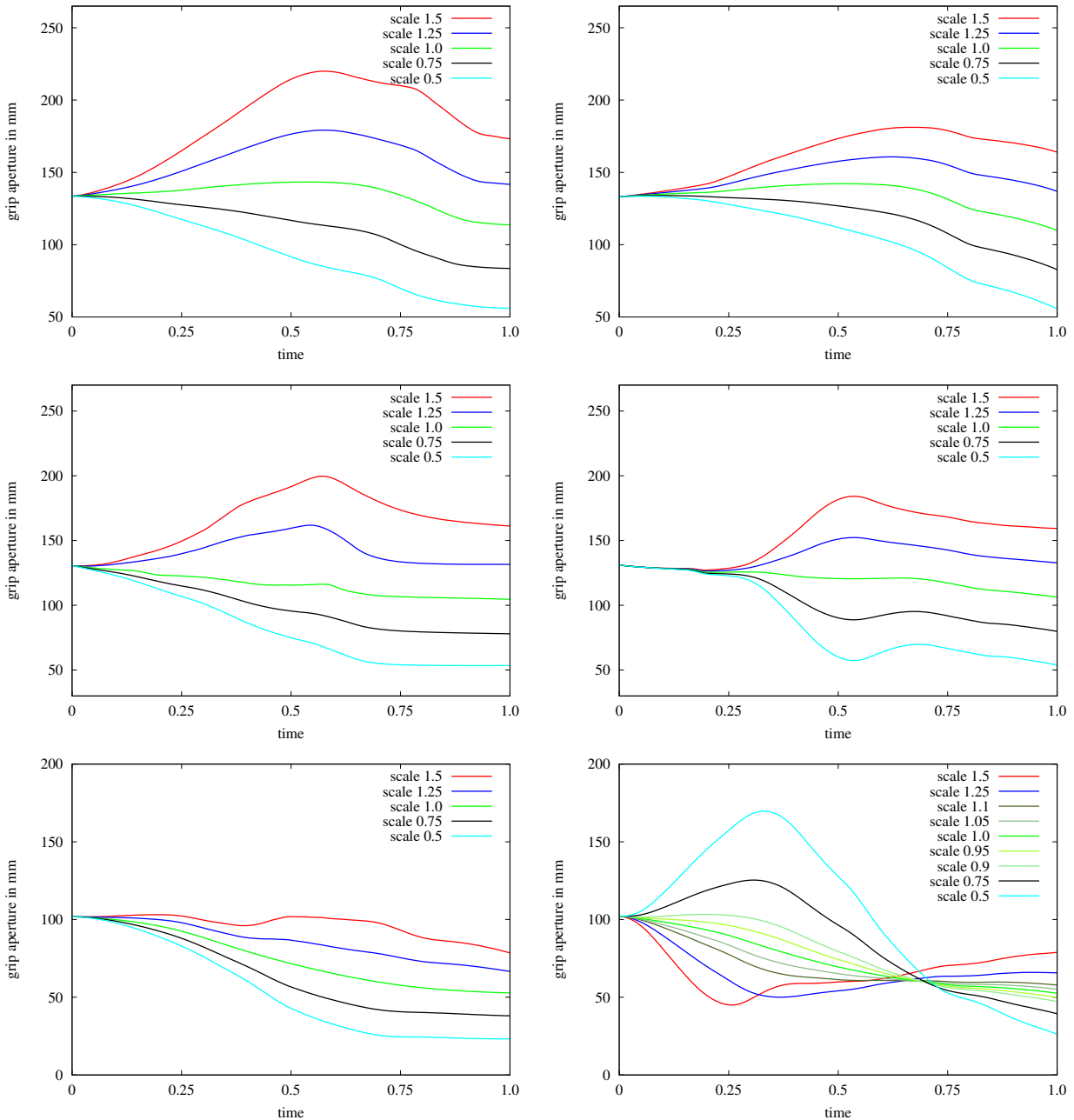


Figure 8.11.: Temporal evolution of the grip aperture movement for different object scales. The left column shows the reproduction of grasp primitives learned using the VSG representation where the right column shows the grip aperture movement for prehensile grasping DMPs. Three different grasp types are depicted: a prismatic wrap grasp (top), a tripod grasp (middle), and a precision grasp (bottom).

a scaling factor  $s_c = u \cdot s$  has been introduced into the VSG representation for the adjustment of the contact springs. In the experiments,  $u = 2$  has been chosen to implement a preshaping behavior similar to the one in human grasping.

As depicted in Figure 8.11, the evolution of the grip aperture has been studied for various object scales. Similar to the findings reported in various human grasp studies such as (Paulignan et al., 1997) and (Castiello, 2005), prehensile fingertip movements have been reproduced which feature a maximum grip aperture that strongly correlates to the object size. An increase in object size led to a larger maximum grip aperture. In Figure 8.11, the grip aperture movements for a prismatic wrap grasp, tripod grasp, and pinch grasp are depicted.

The primitive for prismatic wrap grasp has been derived from a human grasp demonstration with an object with diameter of 114 mm where a maximum grip aperture of 143 mm is featured. Thus, for this specific grasping movement, the ratio  $g = \frac{\text{object diameter}}{\text{max grip aperture}}$  is 0.79 which corresponds to values reported in (Paulignan et al., 1997) for the grasping of larger objects. For increased object scales  $s = 1.5$  and  $s = 1.25$ ,  $g$  remained constant. For smaller objects, a hand preshaping has been imperceptible due to initial open hand configuration which has been larger than the maximum grip aperture in the preshaping phase. However, a change in the slope of the trajectory indicate that, similar to the grasping of larger objects, the hand preshapes at two-thirds of the way. Thus, for  $s = 0.75$  and  $s = 0.5$ , the ratios have been determined to be  $g = 0.77$  and  $g = 0.63$ . As result, a smaller object scale entails a decrease in  $g$ . A similar behavior has been observed when adapting other primitives representing grasps such as a tripod or a pinch grasp to different object sizes.

In comparison to grasp primitives based on the proposed VSG representation, it has been investigated how the DMP-based primitives behave under different object scales. For the prismatic wrap grasp type, a similar grip aperture behavior has been observed. However, the reproduced grasping actions have featured smaller maximum grip apertures. The adaptation to  $s = 1.5$  and  $s = 1.25$  yielded  $g = 0.9$  and  $g = 0.84$ . This is due to the missing encoding of information on intrinsic object properties which might be compensated by define an object-specific scaling of the nonlinear perturbation term. For the tripod grasp, the adaptation to smaller object caused the rapid decrease of the grip aperture. The reason for this behavior lies in the insufficiency of DMPs to adapt to goals which are not in the vicinity of the encoded target. A goal position which changes the direction in which the transformation systems converge to can cause an overshooting of the dynamical system. Thus, moving a contact position in a single dimension along the opposite direction can cause these undesired effects. This deficiency becomes more evident for a pinch grasp which involves only the thumb and the index finger. As it can be seen in the bottom right plot of Figure 8.11, the corresponding DMP could not adapt to larger objects and showed an unexpected behavior for smaller objects. Compared to the DMPs encoding prehensile movements, primitives based on the VSG representation proved to be more adaptive and robust towards different object sizes. The reason for that lies in the explicit encoding of the relation between the grasping embodiment and the object with virtual springs.

### 8.4.3. Behavior under Perturbation

A motivation to propose a representation in the task space is the robust synthesis of movement primitives in dynamic environments. Therefore, in this section, the issue of how the grasp behavior, generated by the proposed VSG representation, changes throughout time with regard to perturbation forces resulting from possible obstacles will be discussed. Due to the dynamical systems formulation, the incorporation of such perturbations is straightforward and is accomplished by adjusting the external force component in Eq. 4.26 accordingly. For a correct evaluation of the results, the behaviors generated by the VSG and the DMP representations have been compared. To simulate a perturbation, a force component which follows a Gaussian distribution and is temporally centered at the end of the preshaping phase is placed to pull the thumb towards the palm.

The results for a spherical precision grasp type are depicted in Figure 8.12 and show that both representations behave robustly under perturbations by attaining the desired grip aperture and converging towards the final grasp configuration without oscillating. In both cases, the perturbations led to a reduction of the grip aperture. For the VSG representation, the perturbation caused a 25% decrease of the maximal grip aperture while 50% decrease has

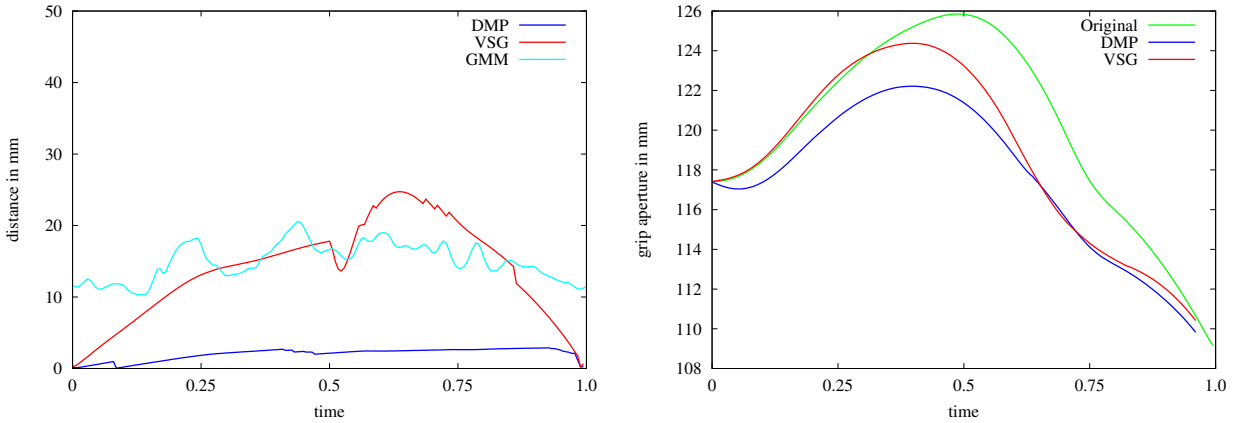


Figure 8.12.: Left: Error between original and reproduced grip aperture movement generated from grasp primitives represented using the VSG, the DMP, and the GMM-based approach. The error is measured as Euclidean distance between the original and reproduced fingertip configuration (involving all five fingers). Right: Grip aperture movement generated under perturbation.

been observed for grasps represented as DMPs. The reason for this difference lies in the virtual finger springs which allow mutual influence between the fingertips, and, thus, help to retain the relative spatial relationships between the fingertips. In contrast to this, the DMP representation treats each fingertip as an independent entity which is decoupled from the remaining fingertips. Therefore, forces exerted on a specific finger do not cause changes in the movements of the other.

## 8.5. Grasp Reproduction

Assuming what grasp type to be executed whether based on information coming from a human operator or by classification of a previously observed the grasping action, the corresponding grasp primitive is selected. The object to be grasped is localized, whereas it is assumed that the object is known and appropriate contact positions are associated with the object model. This has been done manually, but one can think of a grasp planning algorithm in order to obtain this contact information. The object scale is determined based on the distances between the designated contact positions of thumb and virtual finger on the current object and the ones stored in the grasp representation. With the appropriate DMP for the hand approach movement adapted to the object pose, grasp trajectories are generated in the task space.

### 8.5.1. Reproduction of Arm Movements

The movement is mapped to the robot with the method described in Section 7.3.1. For markerless as well as marker-based motion capture data, the proposed method has been evaluated by comparing an inverse kinematics method based on the Jacobian transpose and a one-to-one mapping of the captured joint angles. For the evaluation, the most representative method is to reproduce the movement under similar conditions as captured.

**Marker-based Motion Capture Data** The evaluation of the mapping method has been conducted based on motion data which has been generated within the work of (Stein et al., 2007). These describe various kitchen actions including movements like stirring, cutting, sweeping,



grinding, grating, and pouring. Each arm movement has been captured in the form of trajectories describing the pose of the endeffector in 6D as well as the current arm configuration consisting of 13 DoF joint angle vector (3 DoF for the hip, 1 DoF for sternoclavicular joint, 3 DoF for the shoulder, 2 DoF elbow, 2 DoF for the wrist and 3 DoF for the hand). Based on a predefined mapping table, presumable joint correspondences between the MMM reference model and the robotic embodiment are specified. In the case of the humanoid ARMAR-IIIb, due to the anthropomorphic structure, the arm joint correspondences are obvious whereas the kinematics of the robot do not include the hip pitch and roll joint as well as the sternoclavicular joint. Joints where correspondences can be found are denoted as active joints which are optimized in order to compensate missing joints. The results using of the mapping procedure are illustrated in Figure 8.13a. The left plot of Figure 8.13a shows the joint angle error of a reproduced joint angle configuration on the robot and the reference configuration. Due to redundancy, the inverse kinematics method produces solutions with higher error, while a one-to-one mapping naturally leads to a minimal error. In the center plot of Figure 8.13a, by the right arm endeffector, the deviation of the hand positioning is illustrated. Here, given a endeffector destination, the inverse kinematics method leads to an exact positioning of the endeffector, while using the one-to-one mapping the designated position is not reached. In both plots, it is shown, that the application of the proposed mapping procedure as, a trade-off is attained, which results in a quite accurate endeffector positioning with an maximum error of 25 mm and an acceptable mean joint angle error of 2 degrees for each DoF. One of the most crucial joints which has a huge impact on the style of a trajectory is the shoulder joint. Therefore, the right plot of Figure 8.13a shows the joint angle error for this joint in particular.

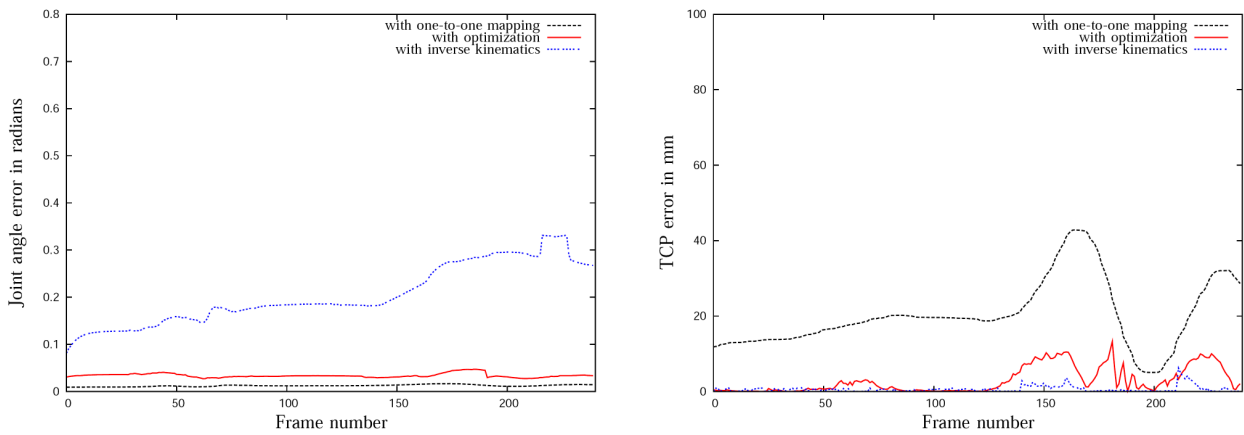
**Markerless Motion Capture Data** For the online reproduction and imitation of observed human motion, the implemented mapping has been evaluated on markerless captured motion data as well. Using the onboard stereo cameras of the ARMAR-IIIb active head upper body movements with a total number of eight DoF were recorded, four for each arm, three DoF for the shoulder joint and one for elbow flexion. Reference points in the task space are denoted as mere 3D hand positions since the orientation cannot be estimated from this motion data. The online reproduction was tested with simple movements like reaching, waiving, and approaching certain postures. Similar to the results previously stated a trade-off between the accuracy of the endeffector position and the joint angle error has been attained. However, due to the reduced number of measured joints, one obtains results with a mean joint angle error of 2.7 degrees for each DoF, as shown in the left plot of Figure 8.13b, and a maximum deviation of 65 mm in the endeffector position of the right arm, as shown in the center plot of Figure 8.13b. The reason for the relatively large deviation is that the utilized vision-based motion capture system is not yet capable of reliably measuring the torso rotation. This lack of information leads to a decreased flexibility throughout the reproduction, assuming the hip joint angles to be fixed. One solution would be to incorporate the hip rotation into the optimization procedure in order to allow for the missing flexibility even if the torso rotation cannot be measured.

### 8.5.2. Reproduction of Finger Movements

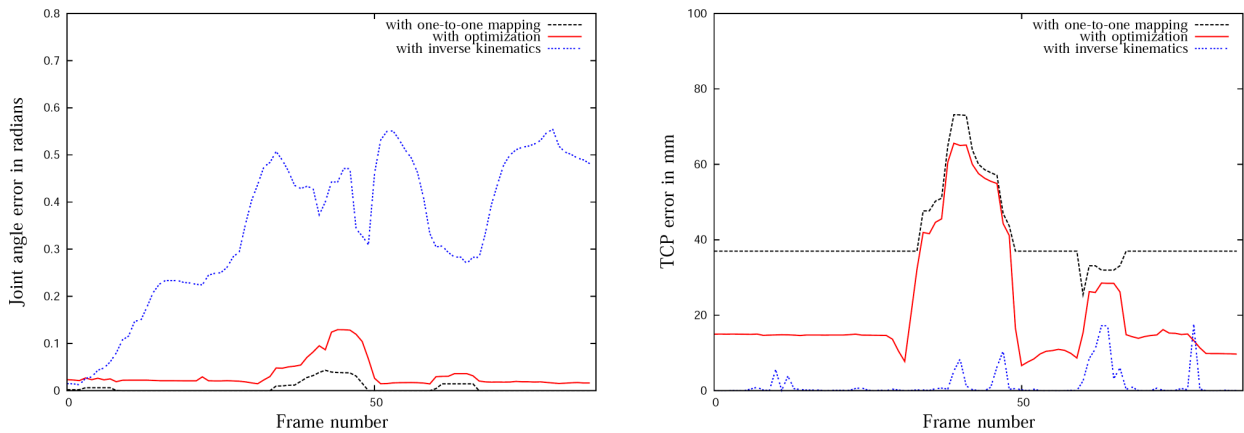
The adaptation of the VSG representation to different robotic hand embodiments is achieved by scaling the stabilization spring lengths according to ratio  $r_R = \frac{l_{H_{mmm}}}{l_{H_R}}$  where  $l_{H_R}$  and  $l_{H_{mmm}}$  stand for the lengths of the robot hand and the MMM hand. Additionally, in order to reproduce a preshaping behavior adapted to the hand, the maximum hand aperture  $m$  is set to  $l_{H_R}$ . The procedure is limited to the mapping to anthropomorphic robot hand systems

featuring five fingers equal to the human hand. For hands with less fingers, similar to (Kang and Ikeuchi, 1997), the fingers are to be grouped to virtual fingers based on manually defined correspondences.

For synthesizing grasps on such hands, the VSG representation has been adapted to the kinematic specifications of the hand system. Furthermore, the springs between the fingertips and the contact springs are adapted to projected contact positions while the object pose has been defined as an attractor for the hand approach DMP. The grasping movement is generated by solving the differential equations incorporated in the DMP and VSG representation using the Runge-Kutta ODE solver of fourth order. For the arm movements, optimized joint angle trajectories are executed whereas the fingers are controlled with IK solutions which are calculated as described in Section 7.3.2. As depicted in Figure 8.13, the reproduction of different grasping actions learned from human demonstrations is simulated for the humanoid robot ARMAR-III. Due to kinematic inaccuracies and possible mapping errors, the generated movement does not allow accurate positioning of the endeffector in the real world. Therefore,



(a) Evaluation results for the reproduction of motion captured by the Vicon system as reported in (Do et al., 2008). Left: Mean joint angle error over all active joints in radians. Center: Deviation of right arm endeffector of the robot and a predefined destination in mm. Right: Mean joint angle error over the shoulder joint in radians.



(b) Evaluation results for the reproduction of vision-based captured motion as reported in (Do et al., 2008). Left: Mean joint angle error over all active joints in radians. Center: Deviation of right arm endeffector of the robot and a predefined destination in mm. Right: Mean joint angle error over the shoulder joint in radians.

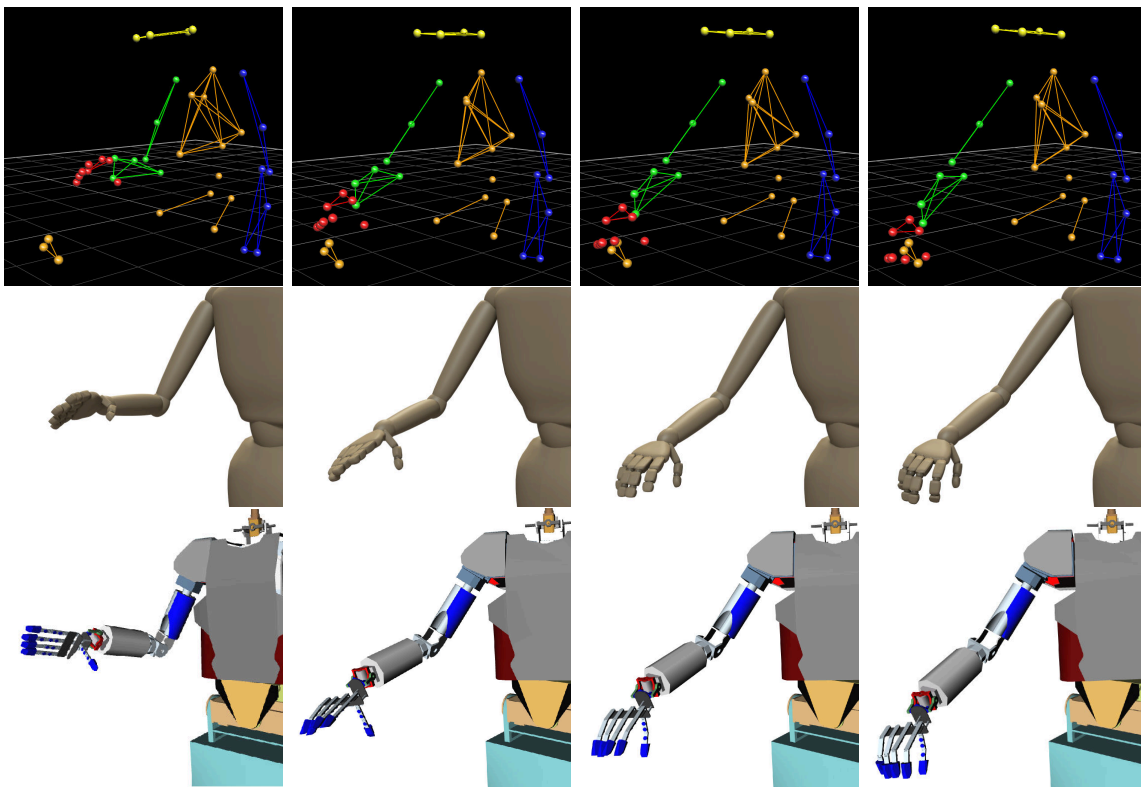


Figure 8.13.: Left: View on the arm configuration observed using the upper body tracking method. Center: Reproduction of the captured arm configuration. Right: Depiction of the optimized joint angle configuration using the MMM interface.

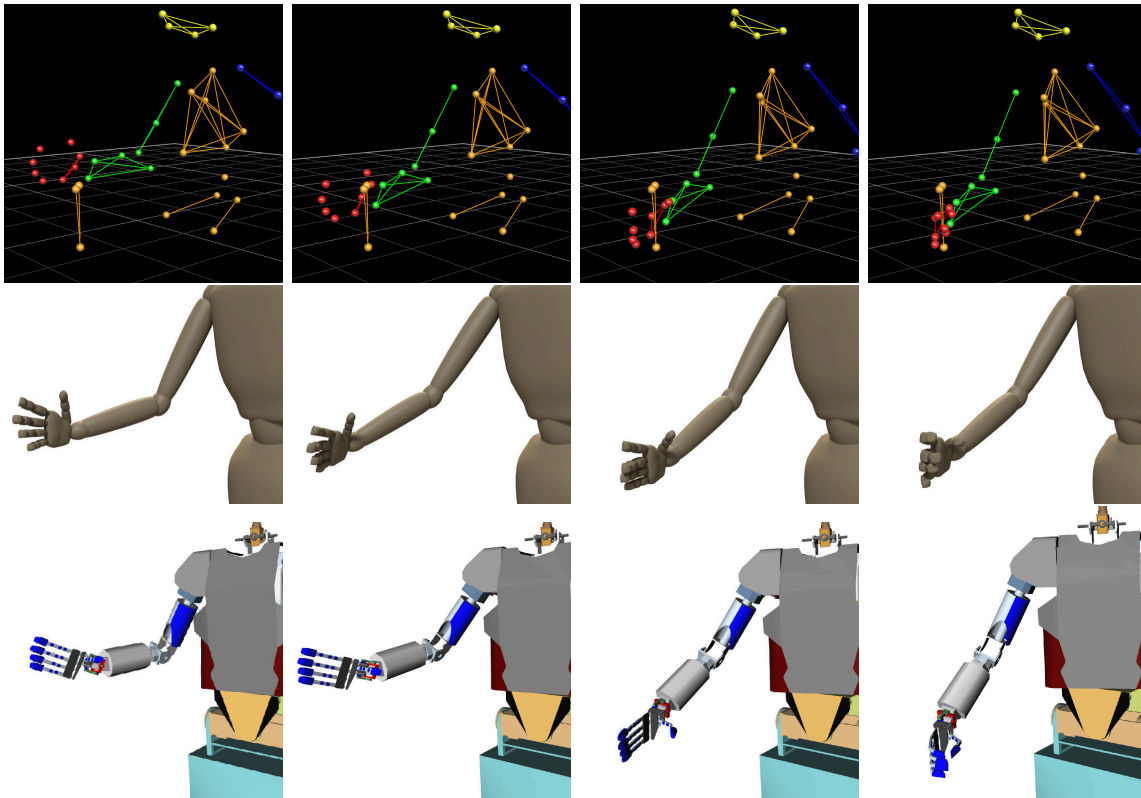
in the last phase of the approach movement, visual servoing is applied in order to align the endeffector with the designated grasp pose. To reproduce prehensile finger movements which accommodate the corrective hand movement, the dynamical system of the VSG representation is perturbed as described in Section 7.5.1. As previously mentioned, in this thesis, grasping movements have been generated and reproduced on the ARMAR-IIIb robot as well as on the Motoman equipped with a Gifu Hand III. The main focus of the experiments conducted on ARMAR-IIIb has been the validity of the proposed approach for learning grasp primitives from human observation. Using the methods introduced in Section 5.3, the robot is endowed with the capability to capture and to process human grasp demonstrations in an online manner using the onboard sensor systems. Due to inaccuracies and occlusion, the distinction between volar and non-volar grasps cannot be made. Thus, the proposed system is constrained to the observation and learning of non-volar grasps. The reproduction of different grasping actions previously learned is depicted in Figure 8.15. The experiments, however, which have been conducted with the Motoman/Gifu system were focused on the representation and reproduction of prehensile movements for various object scales. As depicted in Figure 8.17 and Figure 8.16, a grasp primitive, learned from demonstrations featuring the grasping of an object with a specific shape, is adapted to different object instances which are similarly shaped but with regard to the diameter are differently scaled. Adapted to the grasping hand as well, the grasp primitive is used to generate a grasping action. The multitude of finger joints allows an accurate reproduction of the generated grasping movement. However, the robotic system is very limited in its ability to perform compliant grasping due to an increased sensitivity towards higher forces causing backlashes of the fingers. Therefore, designated contact positions have to be properly approached by the fingers in order to allow a successful grasp. Using the proposed grasp representation, a movement control policy has been derived capable of representing and reproducing prehensile finger movements which are adapted to task- and object-specific constraints as well as the robotic embodiment.

## 8.6. Summary

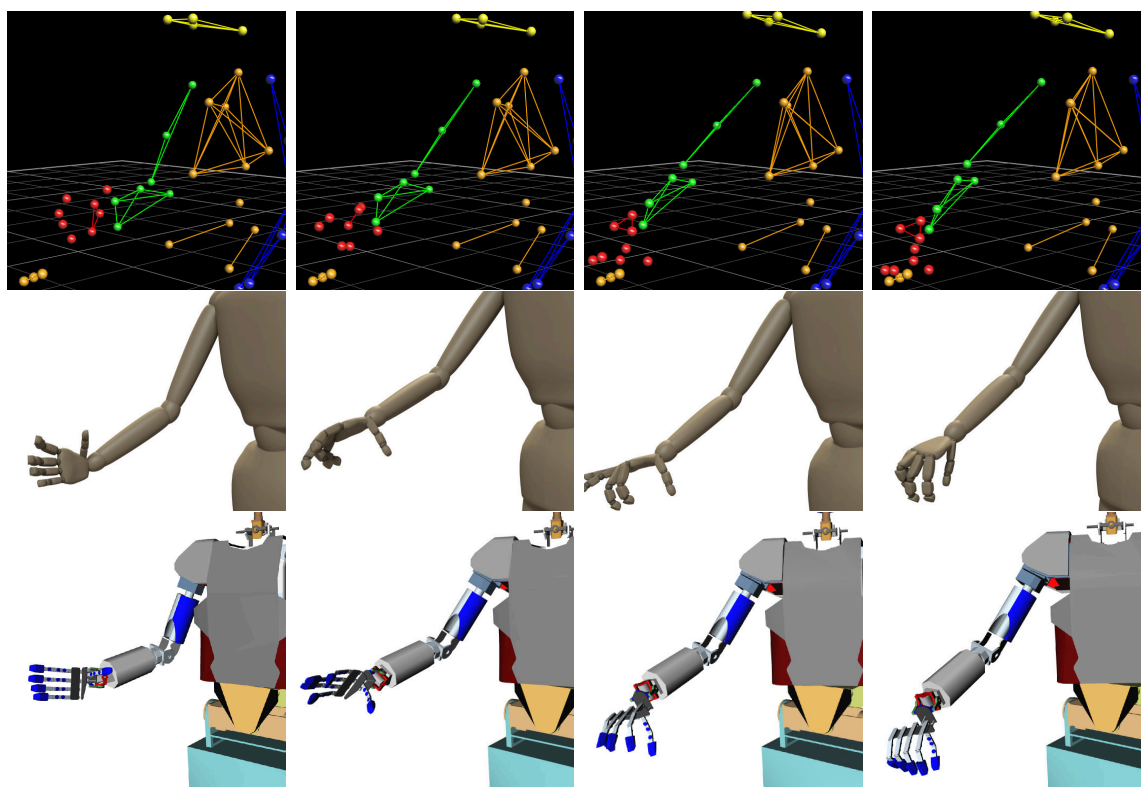
In this chapter, the validity and applicability of the proposed grasp representation for the learning of grasp primitives from human demonstration have been investigated. For this purpose, it has been evaluated how well observed grasping movements can be represented and, how accurate the observed situation has been captured by the components which form the learning framework and, thus, are tailored to proposed grasp representation. Regarding the methods for the observation of human grasp demonstrations, two different approaches have been employed for gathering motion data for the learning of grasp primitives. In particular,



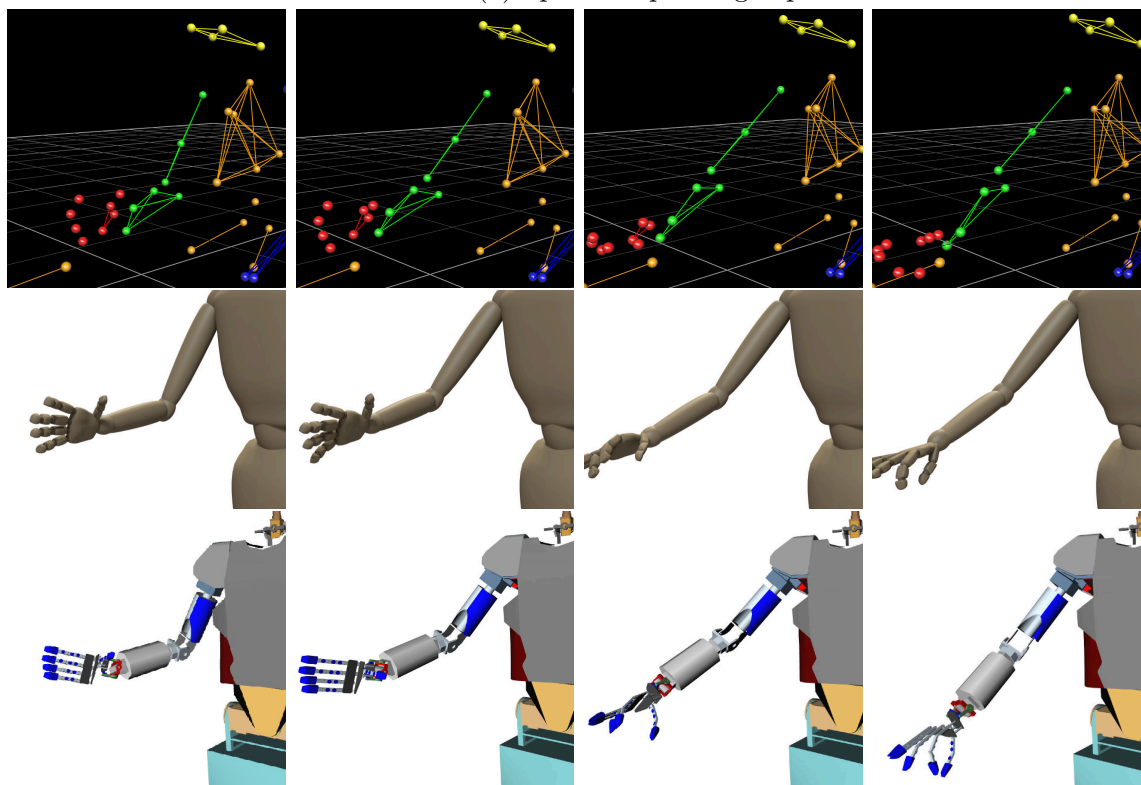
(a) Spherical power grasp



(b) Spherical power grasp



(a) Spherical power grasp



(b) Spherical power grasp

Figure 8.13.: Various grasp primitives learned from marker-based motion capture data. For each grasp, the top line depicts the Vicon motion data. The middle line shows the mapping of the motion to the MMM reference model. The bottom line illustrates the reproduction of the learned grasp types in simulation.

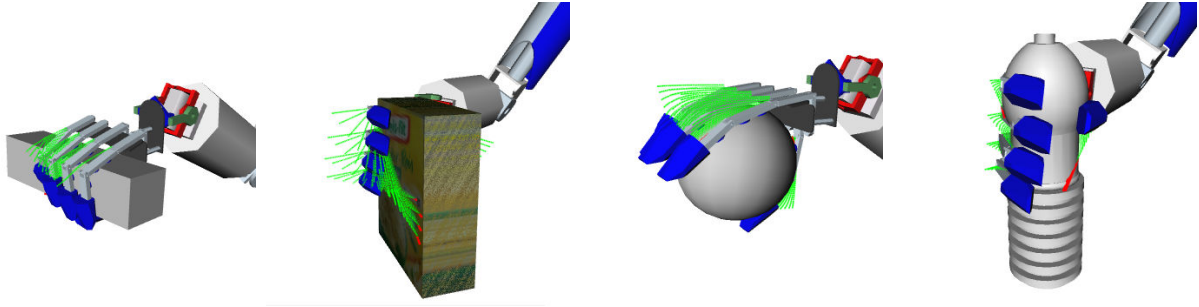


Figure 8.14.: Evolution of Virtual Contact Strip on different object models.

it has been shown that based on observations obtained by the markerless motion capture methods, using the onboard stereo camera system of a humanoid robot, grasp data can be generated which is suitable for the instantiation of the grasp representation. Based on the captured motion data, in comparison to other nonlinear optimization algorithms, the proposed parameter estimation procedure is performed for the efficient identification of model parameters with which the grasp type behavior is described within the proposed representation. The experimental results show that the VSG representation allows the continuous representation of the entire grasping action with a small number of parameters. However, the most important finding is that based on different observations featuring the same behavior the proposed estimation scheme is capable of finding a parameter estimate which uniquely describes the demonstrated grasp type. Based on this finding, it has been shown that the presented framework is capable of learning grasp primitives for grasp types defined in the Cutkosky taxonomy. In order to evaluate the synthesis of the represented grasps, these primitives are adapted to the robotic embodiment as well as different task- and object-specific constraints. Compared to grasping actions represented in the form of a DMP, a formulation which also uses of dynamical systems approach for a continuous description of movements, the proposed approach proved to be more versatile. Due to virtual spring elements which enable the encoding of the finger-object relations not only non-volar grasp could be represented and reproduced, but also volar grasps which require palm contact as well as the wrapping of the fingers around the object in order to maximize the contact areas. Furthermore, it has

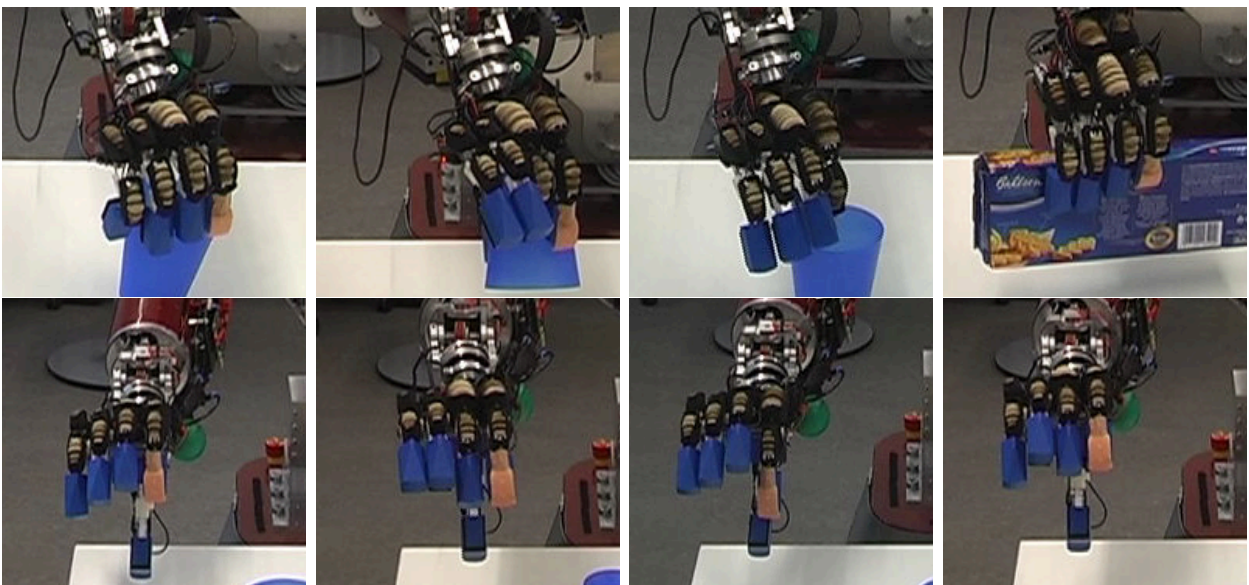


Figure 8.15.: Non-volar grasp types learned from human observation in an online manner.



Figure 8.16.: Grasp preshapes and different scales reproduced on the Motoman/Gifu platform.

be shown that the virtual springs used for the implementation prehensile finger movement synergies contribute to the compactness of the grasp representation and enhance the synthesis of grasping actions under perturbation. The framework has been integrated on different robotic platforms, the humanoid robot ARMAR-IIIb and the Motoman/Gifu platform. For the Motoman/Gifu platform, the adaptation properties of the grasp primitives to different object scales has been studied whereas the framework's capability of learning human grasp behavior in an online manner has been evaluated on ARMAR-IIIb.

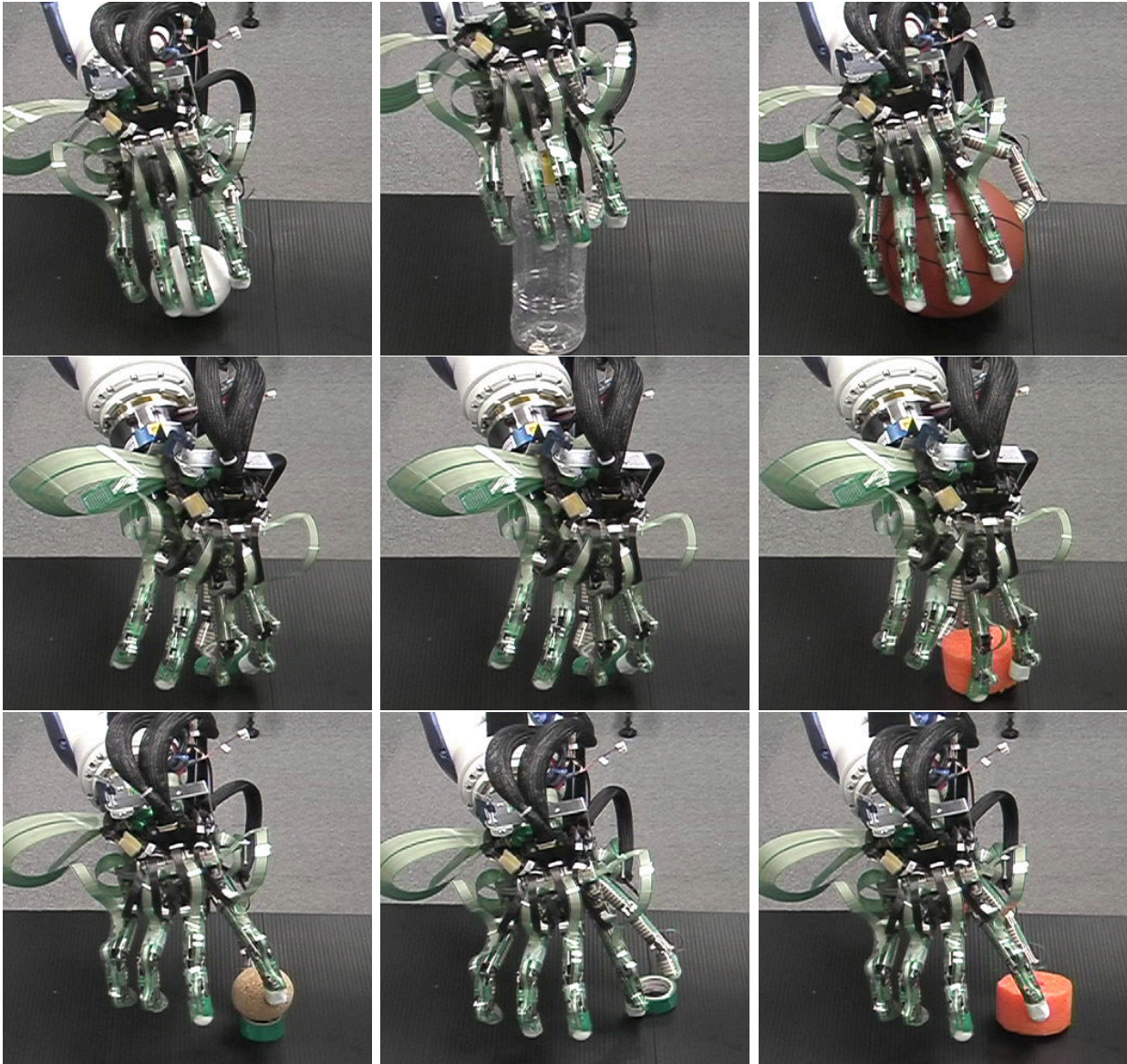


Figure 8.17.: Final grasp postures of reproduced grasp primitives adapted to different object scales and executed by Motoman/Gifu platform.



## 9. Conclusion

The objective of this thesis was to equip humanoid robots with the capabilities to learn grasp primitives from human observation. In this context, the most crucial issue has been the question how grasps can be represented in a comprehensive manner allowing the generalization of observed grasp demonstrations as well as the adaptation of the encoded motor knowledge to versatile task and object requirements. To address this question, the focus of this thesis has been set on the development of a grasp representation and the learning of grasp primitives for the generation of grasping actions applicable in different situations. In this chapter, the contribution of this work is summarized and extensions and future works are discussed.

### 9.1. Contribution

Motivated by the anthropomorphic appearance of humanoid robots and the kinematic similarities between human and humanoids, an intuitive methodology to deal with the complexity of grasping in dynamic environments is to endow robots with a cognitive grasp learning behavior for the acquisition and extension of grasp-related motor knowledge. Following the imitation learning paradigm where actions are learned through observation and generalization of human demonstrations, this thesis presented representations and methods which allow the efficient implementation of such a behavior. They address various aspects such as the capturing of human motion, the representation of the observed motion, the parameter estimation in dynamical systems, and the mapping of motion to humanoid robots. Thus, key scientific contributions of this thesis are summarized as follows:

- **Virtual Spring Grasp Representation**

Grasping is a continuous process which involves multiple phases such as approaching, preshaping, and enclosing the hand. Using discrete grasp representations commonly used in robotic grasping, this process is reduced to a single snapshot of this process causing a loss of valuable information which can facilitate efficient grasping. For a comprehensive representation of grasping actions, in this thesis, the Virtual Spring Grasp representation has been introduced which is based on a dynamical multi-body system in which each fingertip is associated with a mass body. For the modeling of finger movement synergies, which are crucial mechanisms facilitating the coordination of multiple finger movements, mass spring damper system were introduced between these mass bodies. By specifying the stiffness parameters of these springs, a specific grasp-typical behavior has been induced into the representation. The definition of the representation in the task space allows the interpretation of contact points as attractors and, thus, enhances the adaptivity to varying object-specific constraints. To satisfy the requirements of a specific grasping task, the VSG representation has been incorporated in a dynamical system which models the coarse approach movement of the hand.

- **Observation of prehensile fingertip movements**

For the evaluation of the proposed representation, substantial motion data on human grasp demonstrations featuring common grasp types have been collected using a marker-based motion capture system. The recorded grasp sequences have been segmented and

transformed for offline learning of grasp primitives. Towards a cognitive grasp learning behavior, to enable a robot to extend its grasp-related motor knowledge, a method has been presented which allows the detection and the tracking of fingertips in prehensile movements based on foveal views of an active vision system. For the detection of fingertip candidates, steerable filters on multiple scales and a Hough transformation for the extraction of circular image features are used. The tracking of the fingertips is accomplished by transforming the problem into a contour tracking task which is solved using a particle filter algorithm. A refinement of the fingertip position estimates is attained by applying a mean shift procedure. Integrated in an upper body tracking framework, the robot has been enabled to fully observe a grasp demonstration performed by a human.

- **Generation of grasp primitives**

The learning of grasp primitives is mainly accomplished by specifying the characteristic parameters of the observed grasp type. For an efficient parameter estimation, an estimation scheme has been proposed in order to determine the spring constants of the virtual mass spring damper systems which are employed to model the finger movement synergies. The dimensionality and the complexity of the original estimation problem is divided into subproblems. To be able to estimate the descriptive model parameters on observations which are contaminated by noise, a Total Least Squares approach is implemented for the solution of these subproblems. The solutions are combined to a global parameter estimate with which the observed grasp behavior is encoded using the proposed movement representation.

- **Correspondence problem between different kinematics**

For the reproduction of the learned grasp primitives on a humanoid platform, the correspondence problem has been addressed. To enable the generalization and the transfer of observed human motor knowledge to a robotic embodiment, the Master Motor Map (MMM) framework is integrated. Based on a similarity measure and nonlinear optimization methods, a method has been developed which allows the mapping of movements defined in the MMM to the robotic embodiment.

The proposed grasp representation aims towards a simplified description of the continuous grasping process with which the maximum information content extracted from human observation is retained. Therefore, based on the representation, a framework is presented with the purpose of reducing the complexity concerning the acquisition as well as the synthesis of grasp-related motor knowledge. For this purpose, methods have been developed and implemented to enable a humanoid robot to observe, generalize, and reproduce grasping actions which are necessary for an efficient object interaction in human-centered environments. As a result, grasp primitives are obtained which can be adapted to varying task- and object-specific constraints as well as different embodiments with less effort.

## **9.2. Discussion and Outlook**

The representations and methods developed in this thesis address the representation and the synthesis of grasping actions from a perspective which only considers the kinematics of this process. Towards autonomous grasping and manipulation particularly dynamic aspects of grasping have to be addressed in order to increase the robustness and efficiency of the proposed approach. Hence, the future works involves the following tasks:

- **Integration of a grasp planning algorithm**

In this thesis, the evaluation of the proposed grasp representation relies on contact information which has been manually determined. To enable robots to autonomously grasp, the integration of a grasp planning method into the introduced framework is a prerequisite. For this purpose, grasp planning approaches such as (Miller et al., 2003), (Berenson et al., 2007), and (Przybylski et al., 2011) have to be investigated in order to assess the suitability of a possible integration in the proposed learning framework. The combination of both can enhance the grasp planning as well as the learning. From the planning perspective, the rich information encoded in the grasp representation can be exploited to constrain and reduce the search for contact points satisfying dynamic object- and task-specific constraints. On the other hand, contact points calculated by a grasp planning algorithm can be used for prediction and estimation of human-object interaction and, thus, can provide additional information which can be used to enhance the observation as well as the parameter estimation process.

- **Tactile sensing**

Towards robust grasping in dynamic environments tactile sensors have to be considered for contact detection and grasp quality assessment. In order to apply grasp quality measures such as the Wrench Space introduced in (Borst et al., 2004), forces and torques exerted by the fingertips have to be evaluated. In this context, the proposed VSG representation has to be extended with mass bodies which are described by the position and orientation in the task space.

- **Regrasping strategies**

The ability to modify a grasp posture in order to accommodate changing task requirements or unexpected object behavior is an essential part of efficient grasp synthesis. Previous works in this area such as (Vinayavekhin et al., 2011) showed that regrasping strategies can be learned from human observation. Motivated by these approaches, it is desired to extend the proposed approach towards the implementation of a regrasping behavior. In this context, it can be studied how an enriched object representation can be incorporated in the grasp representation and how existing grasp primitives can be used to generate regrasping actions.

- **Learning of manipulation actions**

A future task can be the extension of the grasp representation towards the representation of manipulation actions. The proposed approach can be applied to encode dual arm actions. This is done by interpreting the arm endeffectors as mass bodies and implementing arm movement synergies by means of the virtual spring concept. Furthermore, guided by manipulation taxonomies such as (Bullock and Dollar, 2011), the possibility of using the VSG representation for in-hand manipulation tasks can be investigated. The mapping of complex coordinated finger movements onto virtual springs could be proven advantageous in order to simplify the learning of manipulation actions from human observation.



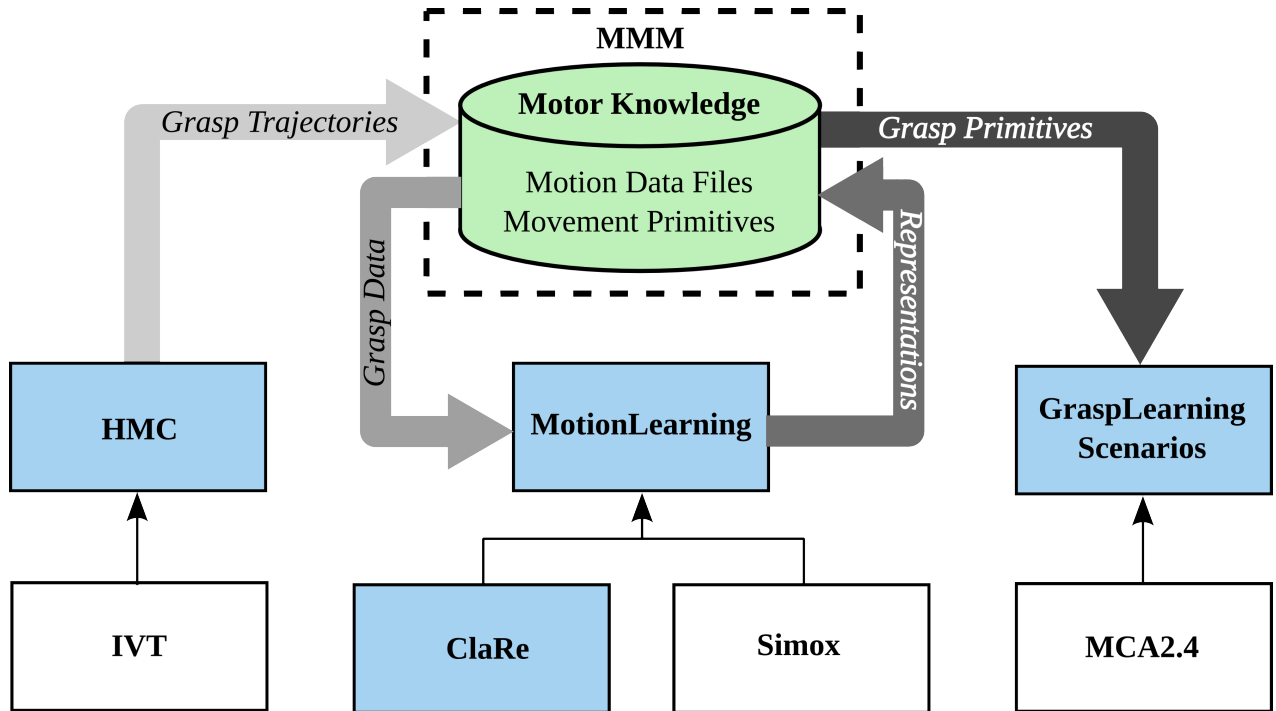


Figure A.1.: Software components which form the grasp learning framework. The blue boxes represent components developed in this thesis. Third-party libraries are depicted as white boxes. The grasp-relevant motor knowledge (primitives as well as motion data) is augmented and accessed via MMM interfaces which are implemented within the HMC, MotionLearning, and the GraspLearning Scenarios. The data flow is illustrated by the thick arrows where the nuances of the arrows indicate different stages of a grasp learning process.

## A. Libraries and Tools

In this appendix, an overview is given on the software tools and libraries which have been developed for the implementation and evaluation of the proposed grasp representation on a robotic platform. For this purpose, a grasp learning framework has been implemented which comprises three main libraries: the *ClaRe* library, the *MotionLearning* library, and the *HMC* library. The abbreviation ClaRe stands for **C**lassification and **R**egression. Thus, the library subsumes various methods which are common in multivariate classification and functional regression with statistical learning methods. Based on these methods, the MotionLearning library provides implementations of learning procedures which are needed for the instantiation of movement representations defined in this library. Since a major focus of this thesis is the learning of movement representations from human demonstrations, methods which allow the capturing of human movements are implemented and wrapped up in the HMC library. In the following, details about the structure and the content of these libraries are given.

### A.1. HMC

The HMC library builds upon the Integrating Vision Toolkit (IVT) (see Azad) and incorporates methods for the markerless capturing of coarse and fine-granular prehensile movements. For this purpose, the HMC library incorporates methods for the tracking of the upper body and the hand. Embedded in the proposed observation scheme, the HMC library is integrated in the ARMAR-III control environment in order to acquire and process the image views which are captured by the stereo camera system of the robot. For the implementation of the tracking procedures, the particle filter framework included by the IVT is used. Running as separate threads, the tracking procedures provide trajectories of the upper body and the fingertips which are passed via the ARMAR-III control framework to a MotionLearning instance for further processing.

As described in Chapter 5, in addition to markerless methods, a marker-based system has been employed in order to obtain motion of human grasp demonstrations. For the marker-based capturing of human motion, the Vicon system provides the NEXUS software package which features methods for the calibration of the motion capture system as well as the recording, the reconstruction, the visualization, and the processing of the captured trajectories. The processed motion data is stored offline. Markerless as well as marker-based captured motion data are streamed respectively loaded into the MotionLearning library for the parameterization of the presented grasp representation.

### A.2. MotionLearning

The MotionLearning library provides abstract methods and classes in order to guide the implementation of procedures for the definition, the learning, and the synthesis of movement representations. For this purpose, the library comprises various base classes which define how concrete classes are to be implemented. For each movement representation, a trajectory class, a learning class, and a control class has to be implemented. The trajectory class converts incoming motion data to a training data set. To facilitate the generation of generalized movement primitives which should be transferable between different embodiments, the trajectory class incorporates an interface to the Master Motor Map in order to enable the normalization of motion data from the specific human embodiment which has been observed. The learning class implements the learning procedure which allows the instantiation of a movement representation based on this data set. Using the control class, the instantiated representation is encoded as a control policy in order to enable the reproduction of the represented action. Focusing on continuous representations of grasping and manipulation actions, the MotionLearning library provides implementation of discrete and periodic arm movements based on Dynamic Movement Primitives as well as the specification of the Virtual Spring Grasp Representation for fine-granular grasping movements.

For the visualization and evaluation of the resulting control policies, a simulation environment provided by the Simox package (see Vahrenkamp) is integrated. Simox comprises libraries and tools for simulation of robotic systems in the course of grasping and manipulation from the kinematics point of view. To enable a human user to generate and inspect different grasp primitives learned from stored grasp examples, the MotionLearning library provides a graphical user interface which is depicted in Figure A.3. The grasp primitives are labeled and stored for later use.

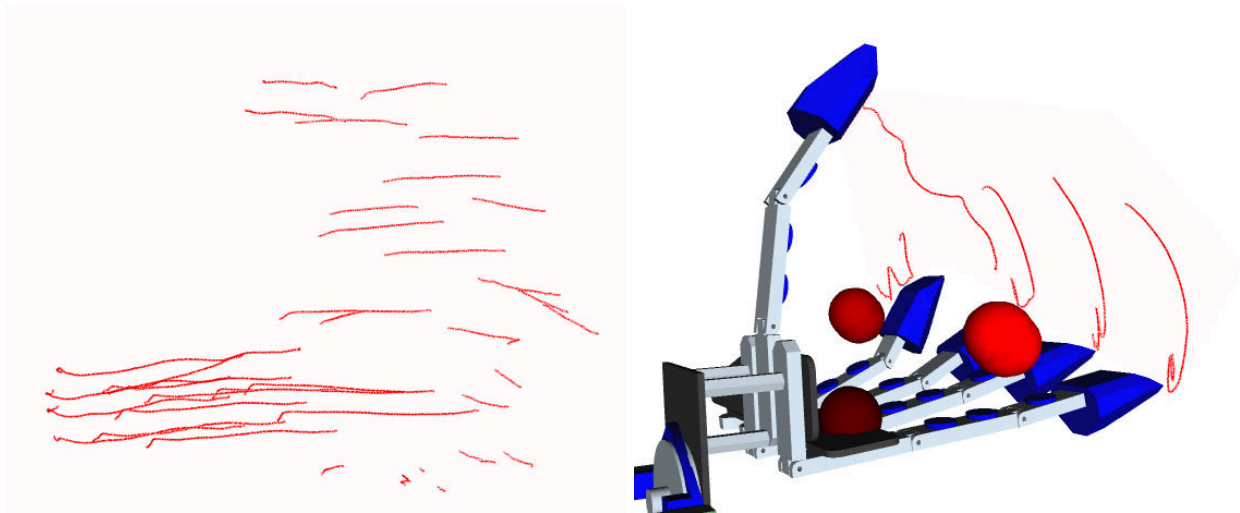


Figure A.2.: Left: Visualization of Vicon marker trajectories representing a prismatic grasp within grasp learning user interface. Right: Corresponding fingertip trajectories transformed into the robot hand coordinate system.

### A.3. ClaRe

The ClaRe library contains various implementations of methods which are needed for the processing of motion data. To facilitate the application of different methods, a data set is uniformly represented in matrix-like structure in which each column corresponds to single data point. The last entry of this column denotes the target value which depending on the objective can be a real value or a discrete class label. For data preparation, various filter methods have been implemented. Furthermore, the ClaRe library incorporates various algorithms such as the Principal Component Analysis, Expectation-Maximization, and Clustering algorithms allowing the spatial and temporal segmentation as well as the dimensionality reduction of motion data. For classification and regression, means are provided for the specification of model parameters with which the relationship between data points and target values can be described. In particular, common data classification such as the Naive Bayesian Classifier, k-Nearest Neighbor, Neural Networks, and Support Vector Machine are incorporated. The regression analysis is accomplished methods using least squares solvers which are tailored to different linear and nonlinear problems. Considering the proposed grasp representation, the Total Least Squares solver which has been outlined in Section 6.3 is of significant importance.

### A.4. Grasp Learning Framework

Based on the functionalities provided by the HMC and MotionLearning libraries, robot skills for the learning and the perception of human grasp demonstrations are implemented and added to the existing skill basis within the robot control software. In combination with already existing skills for the localization of graspable objects, the mapping and execution of generated grasping movements, the grasp observation and learning skills are integrated in grasp learning scenarios in order to ensure the fluent interaction between these components and the correct temporal order of events. In this context, scenarios are software components within the ARMAR-III framework which grant access to the sensor and hardware of the robot as well as all existing motor and perceptual skills implemented on the platform.

The online grasp learning process is initiated by the localization of an object of interest in the scene. Using the object localization methods implemented on the robot, a mode-based

approach is applied to recognize known single-colored objects where textured objects are localized with a data-driven approach using SIFT features (see (Azad et al., 2009)). The upper body tracking thread which starts capturing the human motion when the head and the hands are localized based on a skin color cue. The upper body tracking is stopped once one of the hands reaches the object of interest. Human hand and finger movements are recorded using the markerless methods presented in Chapter 5. The trajectories representing the entire grasping action and additional information of the demonstrated grasp type (e.g. number of involved fingers, palm contact) which originates from human feedback are streamed to a MotionLearning instance for the instantiation of a grasp primitive.

To derive grasp primitives from Vicon motion data, the graphical user interface provided by the MotionLearning library offers allows the loading, visualization, and processing of Vicon motion data files. As depicted in Figure A.2, raw marker trajectories as well as the transformed grasp data can be visualized within the interface. For the parameterization of the proposed grasp representation, the interface provides access to the Total Least Squares solver. Once the parameter estimation process has been performed, the resulting grasp primitive is saved to a file along with additional grasp information.

For the reproduction of learned grasping action, the corresponding primitive can be read either from a stream coming from the MotionLearning library or from an offline stored file. With the read grasp parameters a control policy is inferred based on the dynamical system incorporated proposed grasp representation. The encoded action is generated by applying the ordinary differential equations solver implemented in the GSL library using a fourth-order Runge-Kutta stepping function. The generated movement is mapped according the method described in Section 7.3 and executed as described in Section 7.5.



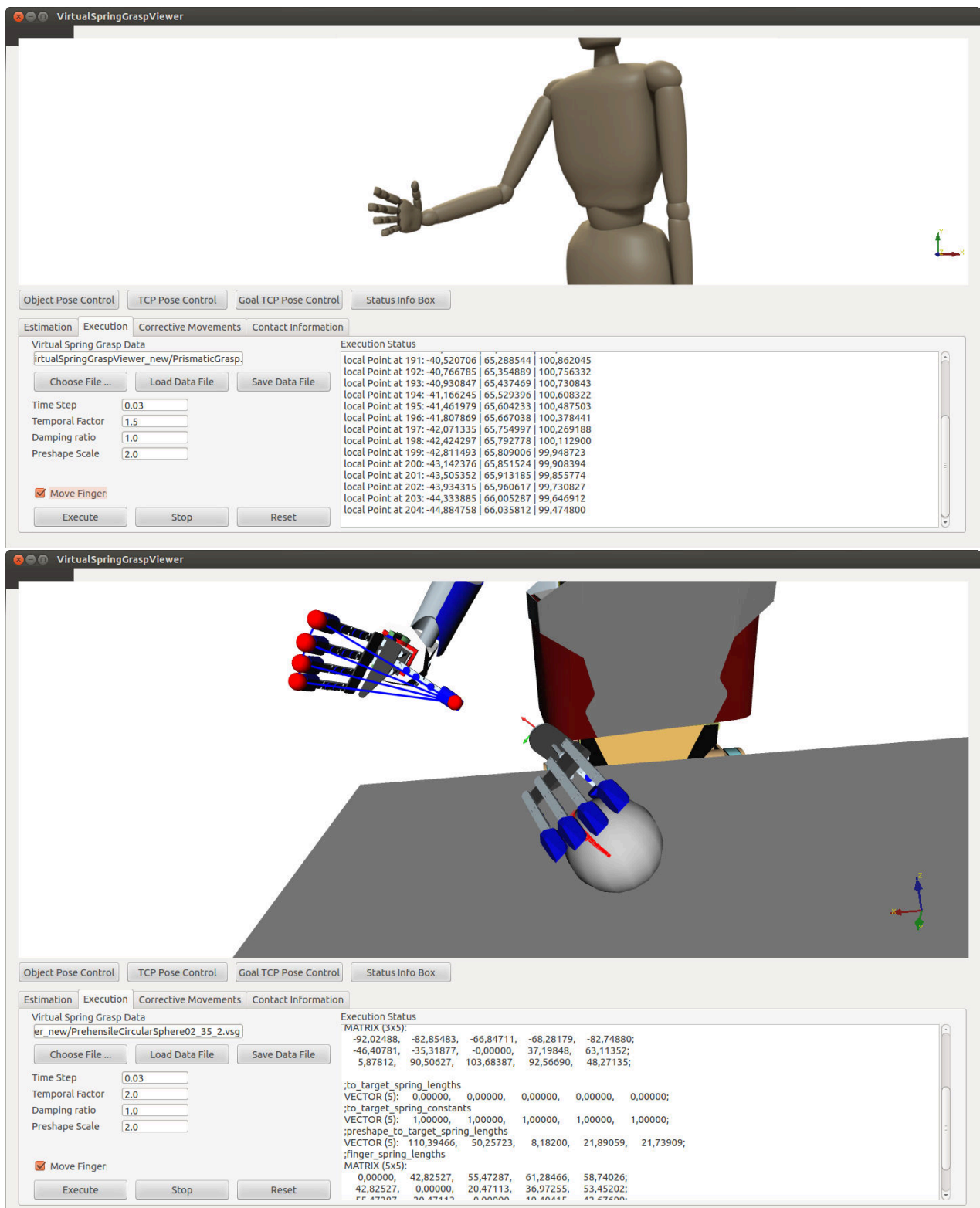


Figure A.3.: Snapshots of the graphical user interface of the grasp learning framework. Top: Interface for the learning of grasp primitives using motion data mapped on the MMM model. Bottom: Interface for the reproduction and simulation of grasp primitives using the ARMAR-III model.



## B. Learning of Rhythmic Manipulation Actions

So far, the representations presented in this thesis are suitable for the learning of discrete actions. However, rhythmic actions such as walking, stirring, and wiping are as essential as discrete actions in human motor control. Unlike in discrete actions where the endeffector is directed towards a target and the action is considered to be finished once the designated target has been reached, in rhythmic actions, the endeffector is supposed to repeat a periodic motion pattern indefinitely. The evidence that rhythmic actions are not a composition of discrete segments has been provided by (Schaal et al., 2004) where fMRI studies showed that different brain areas are activated. Thus, in order to suitably encode and reproduce a rhythmic action, a different representation has to be defined. Addressing this issue, in the following sections, it is discussed how a representation for periodic movements can be formed based on the methods which have been discussed in Chapter 4.

### B.1. Representation of Rhythmic Actions

As previously discussed in Section B.1, the continuous representation of discrete endeffector movements has been addressed in numerous works, specifically in the context of imitation learning. Representations for periodic motion patterns have been studied with regard to biped locomotion of humanoid robots. In particular, Central Pattern Generators describing neural networks which are responsible for the generation of periodic motion patterns in most biological organisms have been of great interest. On this view, (Kimura et al., 2007), (Ijspeert, 2008), and (Righetti and Ijspeert, 2006) have introduced models based dynamical systems in order to implement such mechanisms on robots. In a similar fashion, as described in (Ijspeert et al., 2002), the DMP representation which has been introduced in Section 4.4 and used for the description of discrete arm reaching movements can be modified in order enable the representation of periodic movements as well. As described in Section 4.4.1, the main components of a DMP are a canonical system, a transformation system, and perturbation term. For the encoding of a periodic movement, instead of implementing the canonical system as point attractor, a limit cycle attractor is used to induce an indefinite repetitive behavior into the dynamical system. Driven by a phase variable  $\phi = [0, 2\pi]$ , the corresponding canonical system is defined as follows:

$$\tau_p \dot{\phi} = 1 \quad (\text{B.1})$$

$$\tau_p \dot{r} = k_p(a - r) \quad (\text{B.2})$$

with  $\tau_p = \frac{T_p}{2\pi}$  denoting the duration of the each periodic cycle. The current amplitude of the system is described by  $r$  whose convergence to a desired value  $a$  is ensured by Eq. B.2. The convergence behavior is defined by  $k_p$ . The transformation systems are defined as in discrete DMPs where the target position  $\mathbf{x}_g$  represents the center around which the system oscillates. In the following, it is referred to  $\mathbf{x}_g$  as the anchor position of a periodic pattern. The force term  $\mathbf{f}_p$  which perturbs the system can be written as follows:

$$\mathbf{f}_p(\phi, r) = \frac{\sum_{i=1}^N \varphi_i(\phi) w_{p,i} \mathbf{x}_p}{\sum_{i=1}^N \varphi_i(\phi)} \quad (\text{B.3})$$

with  $\varphi_i(\phi) = \exp\left(-\frac{(\text{mod}(\phi, 2\pi) - c_i)^2}{2\sigma_i}\right)$  being the basis function centered at  $c_i$  and with variance  $\sigma_i$ . The vector  $\mathbf{x}_p = (r \cos \phi, r \sin \phi)^T$  determines the direction of  $f_p$ . By adjusting the weights  $w_{p,i}$ , the perturbation term can be adapted to a demonstrated trajectory. The resulting dynamical system can be adapted to different conditions such as changing anchor positions and varying amplitude and frequency values.

### B.1.1. Encoding of Transient Periodic Movements

Commonly, most rhythmic actions start with a discrete movement. For instance, an initial step based on different conditions has to be made before a walking pattern can be repetitively executed. In the context of wiping, a discrete approach movement which has to be performed in order to center the periodic motion pattern at a specific location blends into a repetitive wiping motion. Thus, for a complete description of a rhythmic action, a generalized movement representation has to be formalized which is capable of representing both, the periodic as well as the discrete part of the movement to be encoded. In the following, the discrete part of a rhythmic action is referred to as a transient.

Based on (Ijspeert et al., 2002) where a periodic pattern is represented as an asymptotically stable limit cycle using the DMP approach, in (Ernesti et al., 2012), an extension is suggested which allows the representation of a periodic motion as well as its corresponding discrete transient movement. Transients are represented as trajectories converging towards the limit cycle. The encoding of both, periodic and transient motion, is accomplished by introducing a two-dimensional canonical system in the DMP formulation. In addition to the phase variable  $\phi$ ,  $r$  is introduced to describe distance from the periodic pattern. This yields the state of the DMP  $s(t) := (\phi(t), r(t))$  as the solution  $(\phi, r)$  of the following ordinary differential equation:

$$\tau_p \dot{s} \begin{cases} \tau_p \dot{\phi} & = 1 \\ \tau_p \dot{r} & = \eta(a^\alpha - r^\alpha)r^\beta. \end{cases} \quad (\text{B.4})$$

The amplitude  $a > 0$  also denotes the radius of the limit cycle. The constants  $\eta, \alpha, \beta > 0$  are variables which allow to adjust the convergence behavior. The value of  $\phi$  is linearly increasing whereas  $r$  converges monotonously to  $a$ . Thus, by interpreting  $(\phi, r)$  as polar coordinates the solution of B.4 converges towards a circle with radius  $a$  around the origin on the phase plane. To encode a demonstrated wiping action described by  $\mathbf{X}_w, \dot{\mathbf{X}}_w, \ddot{\mathbf{X}}_w \in \mathbb{R}^{D \times T_e}$ , a transformation system in the form of  $D$  critically-damped spring systems is defined which initially converges towards a global point attractor  $\mathbf{x}_g \in \mathbb{R}^D$  and continues to oscillate around  $\mathbf{x}_g$ . The transformation system is specified as follows:

$$\tau_p \dot{\mathbf{v}} = k_w(\mathbf{x}_g - \mathbf{x}) - \zeta_w \mathbf{v} + \mathbf{f}_w(\phi, r), \quad (\text{B.5})$$

$$\tau_p \dot{\mathbf{x}} = \mathbf{v}. \quad (\text{B.6})$$

The constants  $k_w$  and  $\zeta_w$  are chosen in order to create a critically-damped system. Due to the two-dimensional state space, the perturbation force is composed of two different terms representing the periodic motion pattern as well as its corresponding transient. Thus, with  $M > 0$  basis functions for the encoding of periodic characteristics and  $N > 0$  for the transient part,  $\mathbf{f}_w$  is defined as follows:

$$\mathbf{f}_w(\phi, r) = \frac{\sum_{j=1}^M \tilde{\psi}_j(\phi, r) w_{t,j} + \sum_{i=1}^N \tilde{\varphi}_i(\phi, r) w_{p,i}}{\sum_{j=1}^M \tilde{\psi}_j(\phi, r) + \sum_{i=1}^N \tilde{\varphi}_i(\phi, r)}, \quad (\text{B.7})$$

where  $\mathbf{W} = (w_{p,1}, \dots, w_{p,N}, w_{t,1}, \dots, w_{t,M})^T \in \mathbb{R}^{N+M}$  contains the weights which can be adjusted to fit the desired trajectory  $(\mathbf{X}_w, \dot{\mathbf{X}}_w, \ddot{\mathbf{X}}_w)$ . The basis functions  $\tilde{\psi}_j$  encode the transient part

of the motion while the periodic part is modeled using  $\tilde{\varphi}_i$ . In addition to the previously introduced basis functions,  $\tilde{\psi}_j$  and  $\tilde{\varphi}_i$  incorporate a term which determines the influence of these functions close to the limit cycle. To implement a fading behavior from the transient to the periodic movement  $b_p(r)$  and  $b_t(r)$  are introduced as follows:

$$\tilde{\psi}_j(\phi, r) = b_p(r)\psi_j(\phi) \quad (\text{B.8})$$

$$\tilde{\varphi}_i(\phi, r) = b_t(r)\varphi_i(\phi, r), \quad (\text{B.9})$$

where  $b_t(r)$  causes  $\varphi_i$  to vanish close to the limit cycle, while  $b_p(r)$  diminishes the influence of  $\psi_j$  the farther the systems moves away from the limit cycle. Therefore,  $b_p(r)$  and  $b_t(r)$  can be defined as follows:

$$b_p(r) = \begin{cases} \exp(-\nu(r-a)^2) & r \in (a, \infty) \\ 1 & r \in (0, a] \end{cases} \quad (\text{B.10})$$

$$b_t(r) = \begin{cases} 1 & r \in [a, \infty) \\ \exp(-\nu(r-a)^2) & r \in (0, a), \end{cases} \quad (\text{B.11})$$

where the constant  $\nu > 0$  is used to control the fading behavior.

### B.1.2. Learning of Rhythmic Actions

Based on the previously described DMP formulation, a movement primitive for a wiping action can be learned. The learning of a wiping action is decoupled in two phases: the encoding of a human wiping demonstration and the adaptation of the resulting wiping movement primitive to the surface to be wiped. Motion data of human wiping demonstrations have been recorded using the Vicon system previously described in Section 5.2. For this purpose, reflective markers have been placed on the hand and the wrist of the human subject allowing the capturing of trajectories which describe the hand's wiping movements in the task space. In this thesis, four styles of wiping movements were investigated which feature varying discrete initial approach movements (simple and complex) and different periodic wiping patterns (circle and figure eight). Each demonstration is segmented in order to identify the transient part and the periodic pattern of a wiping action. To encode the characteristics of the demonstrated wiping movement, the weights in Eq. B.7 are determined using linear regression techniques as introduced in Section 6.5. Initially, the wiping movement demonstrated in the task space is learned in the  $(x, y)$ -plane disregarding the surface contact which yields a wiping DMP with a two-dimensional transformation system. The captured human demonstrations and the generated trajectories using the correlating DMP are illustrated in Figure B.1.

### Segmentation of a Wiping Movement

Before a wiping movement demonstrated in the  $(x, y)$ -plane can be represented as using the DMP formulation described in Section B.1.1, for a recorded trajectory  $\mathbf{X}_w = \{\mathbf{x}_w(t)\}$ ,  $\mathbf{x}_w(t) \in \mathbb{R}^2$  with  $T_s \leq t \leq T_e$ , the transient fading time  $T_{trans}$  and the anchor point  $\mathbf{x}_g$  has to be determined. For that reason, the trajectory is segmented into two parts, i.e.  $\mathbf{X}_{trans} = (\mathbf{x}_w(T_s), \dots, \mathbf{x}_w(T_{trans}))$  and  $\mathbf{X}_p = (\mathbf{x}_w(T_c^-), \dots, \mathbf{x}_w(T_c^+))$  with  $T_{trans} < T_c^- < T_c^+ \leq T_e$  denoting the start and the end of a periodic cycle. The segment  $\mathbf{X}_{trans}$  contains the data points corresponding to the transient part where  $\mathbf{X}_p$  is the vector of data points corresponding to the periodic pattern. The segmentation is based on the assumption that in each cycle the vector  $\mathbf{v}(t) = \mathbf{x}_p^o(t) - \mathbf{x}_p(t)$  between a trajectory point  $\mathbf{x}_p(t) \in \mathbf{X}_p$  and the opposing point  $\mathbf{x}_p^o(t) \in \mathbf{X}_p$

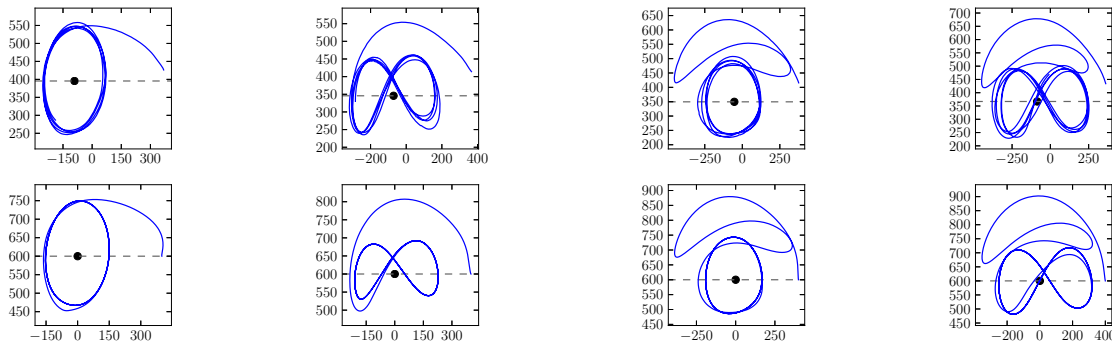


Figure B.1.: From the left to right plot all recorded wiping styles (1,2,3,4) are shown. The top line shows the original motion that was captured from human demonstration. On the bottom, the trajectories are visualized which have been generated by the corresponding DMP adapted to the different starting and anchor points as indicated by the dashed lines.

remains unchanged throughout the entire periodic movement. For each point  $\mathbf{x}_w(t)$ , the opposing point  $\mathbf{x}_w^o(t)$  is determined from a set of points  $\mathbf{X}_w^o(t)$  according to following rule:

$$\mathbf{x}_w^o(t) = \underset{\mathbf{x} \in \mathbf{X}_w^o(t)}{\operatorname{argmin}} \|\mathbf{x}_w(t) - \mathbf{x}\|,$$

where  $\mathbf{X}_w^o(t)$  originates from the intersection of a plane at  $\mathbf{x}_w(t)$  and a normal vector  $\mathbf{n} = \mathbf{x}_w(t+1) - \mathbf{x}_w(t-1)$  with  $\mathbf{X}_w^+ = \{\mathbf{x}_w(t^+)\}$ ,  $t < t^+ \leq T_e$ . The transition point  $T_{trans} = t_1$  from discrete part to periodic part is found if  $v_{t_1} = v_{t_2}$  for  $T_s < t_1 < t_2 < T_e$ . For the sake of robustness, in every step, this check is performed for a sequence of points  $x_w(t-5), \dots, x_w(t-5)$ . Once  $T_{trans}$  is found, for the first periodic cycle, the duration of a cycle is determined by  $T_p = T_c^+ - T_c^-$  with  $T_c^- = T_{trans}$  and  $T_c^+ = t_2$  and, thus, the anchor point of the periodic movement is calculated as follows:

$$\mathbf{x}_g = \frac{1}{T_p} \sum_{t=T_{trans}+1}^{T_{trans}+T_p} \mathbf{x}_w(t).$$

### B.1.3. Adaptation to Environment

Due to inaccuracies within the kinematic chain from platform to the robot's endeffector, contact with the surface to be wiped and the wiping tool cannot be guaranteed. To attain a goal-directed reproduction of the learned wiping action, the force torque sensor is exploited in order to adapt the movement of the endeffector to the surface shape during the execution of the generated wiping movement. Following the force profile adaptation method introduced in (Gams et al., 2010), a force-feedback control mechanism is implemented which moves the endeffector towards the surface while executing the wiping pattern. For an endeffector whose current pose is described by  $\mathbf{x}(t) \in \mathbb{R}^6$ , the velocity during the execution of a wiping action is calculated by:

$$\mathbf{v}(t) = \mathbf{v}_w(t) + \mathbf{K}_w \mathbf{S}_w (\mathbf{f}_m(t) - \mathbf{f}_o), \quad (\text{B.12})$$

where  $\mathbf{v}_w(t)$  denotes the desired velocities induced by the DMP control policy.  $\mathbf{S}_w \in \mathbb{R}^{6 \times 6}$  stands for the force selection matrix and determines in which force direction the policy is to be adapted. The adaptation behavior is determined by the force gain matrix  $\mathbf{K}_w \in \mathbb{R}^{6 \times 6}$ .  $\mathbf{f}_m(t)$  denote the current force measurements whereas  $\mathbf{f}_o$  describes the force state when the robot is not in contact with the environment. For a flat surface, Eq. B.12 is specified with  $\mathbf{K}_w = \operatorname{diag}(0, 0, k_z, 0, 0, 0)$  and  $\mathbf{S}_w = \operatorname{diag}(0, 0, 1, 0, 0, 0)$  yielding a force-based adaptation in the

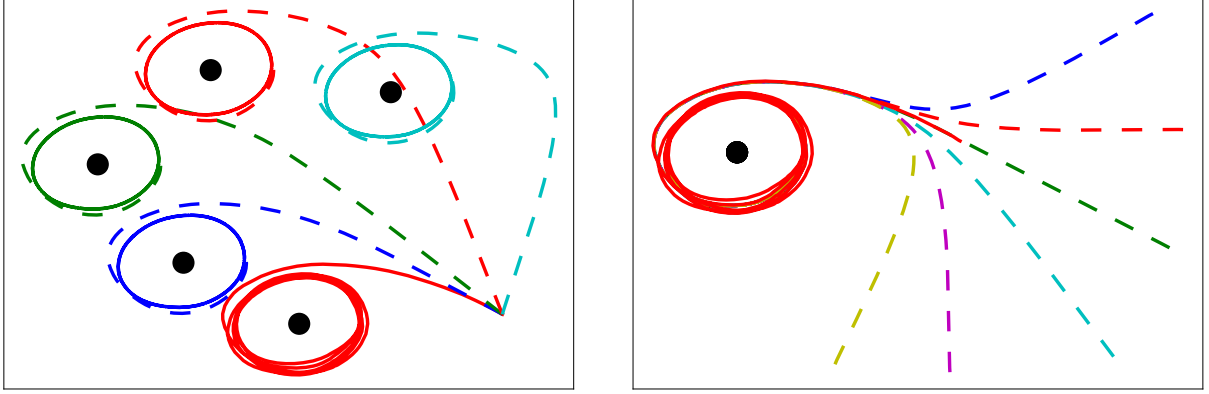


Figure B.2.: On the left the generalization to different anchor points (black dots) is visualized. The right plot shows runs with different starting points of the same motion. All dashed plots show reproductions of wiping style 1 and the solid line corresponds to the original trajectory.

$z$ -direction which changes the height of the endeffector in platform coordinates. Based on presumed table height  $z_0$ , the desired movement is calculated in discrete time steps:

$$\dot{z}(t) = k_z(f_z(t) - f_{0,z}), \quad (\text{B.13})$$

$$z(t) = z_0 + \dot{z}\Delta t \quad (\text{B.14})$$

$$(\text{B.15})$$

where  $f_z(t)$  is the measured force in the  $z$ -direction and  $f_{0,z}$  is the desired force with which the robot should press on the surface. However, reliable force measurements in a single direction can be only obtained from constant movement or permanent contact with the surface. To circumvent these restrictions, the problem is simplified by using the length of the force vector  $\mathbf{f}_m = \sqrt{f_x^2 + f_y^2 + f_z^2}$  instead of  $f_z$ . This simplification causes the robot move upwards every time it hits something. As previously mentioned,  $k_z$  controls the adaptation behavior of the movement and should be chosen in a way that on the one hand the upward movement does not get too large and on the other hand enables the robot to quickly adapt to contact forces. Through experiments the parameters have been empirically set to  $k_z = 10 \frac{\text{kg}}{\text{s}}$  and  $f_{0,z} = 15\text{N}$ . Based on the force-based surface adaptation a trajectory  $z_w(t)$  is recorded for every periodic cycle. The wiping DMP is repeatedly reproduced until the force torque measurements during the execution of  $z_w(t)$  meet predefined constraints which guarantee that the endeffector applies the desired pressure on the surface to be wiped. Based on  $z_w(t)$  an additional transformation system is learned in order to augment the existing DMP.

## Reproduction of a Wiping Movement

For each wiping style, a DMP has been learned and, hence, each DMP encodes a single approach movement and a periodic movement. For the representation of the demonstrated movements approximately  $M = N = 40$  basis functions have been employed. In order to synthesize the learned wiping movement on the robot, the DMP is parameterized with scene-specific start and anchor position. The capability of this DMP formulation regarding the generalization to different start and anchor positions is shown in B.2. To demonstrate the adaptability of the used representation, for each DMP, wiping movements were generated starting from three points and leading to three different goal positions which are equivalent

to the anchor points of the periodic pattern. By integrating the resulting control policy and using differential kinematics the wiping movements could be reproduced on the humanoid platform ARMAR-IIIb. To enable the force adaptation in an online manner, the reproduction speed has to be reduced. Hence, the wiping movements have been reproduced at half of the demonstrated speed. In order to visualize the executed robot trajectories, a pen is attached to the sponge. To minimize the friction between the sponge and the surface, only soft pressure is applied during the wiping. Hence, contact is already determined if only parts of the sponge, not necessarily the pen, touch the surface leading to discontinuities in the drawn trajectories. The results of the experiments are depicted in Figure B.3.

### **B.2. Conclusion**

The DMP formulation which is described in Section 4.4 combines the representation of discrete as well as periodic movements in a single unit. Thus, a generalized representation is provided which facilitates the learning and reproduction of primitives for the encoding of rhythmic action and its potential transient behavior. Experiments in the context of the table wiping task showed the validity of this approach. Using this DMP formulation, adaptive movement primitives have been learned from human demonstrations. Using these learned primitives, wiping actions adapted to different start and goal conditions could be reproduced on a humanoid platform. In addition, a method is described which allows the augmentation of the primitives to accommodate environmental features. Based on an active-compliant approach, force torque sensor measurements are used to determine an adaptation of the primitive in an online manner.



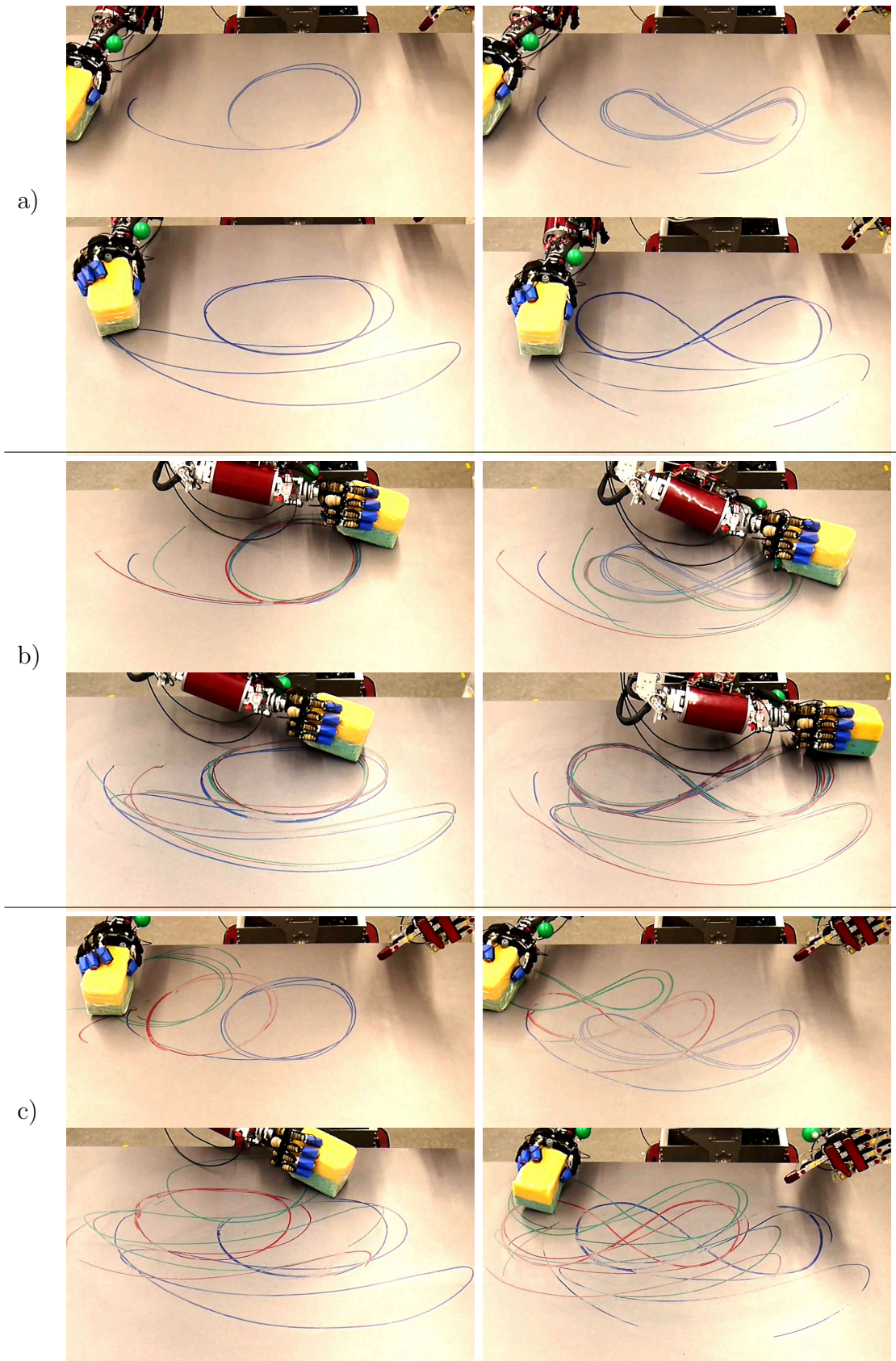


Figure B.3.: a) Single reproduction of a wiping movement from style 1, 2, 3, and 4. The trajectory is drawn using a pen attached to the sponge. b) Multiple reproductions of a wiping movement with different start positions. c) Multiple reproductions of a wiping movement with different goal positions (equivalent to the anchor points of the periodic pattern).



## C. Reasoning of an Action based on Object-Action Affordances

The Dynamic Movement Primitive which has been learned to encode a demonstrated wiping movement has been adapted and augmented in order to meet specific requirements imposed by the current wiping object and the environment. In the following sections, mechanisms are discussed which allow the continuous grounding, adaptation, and augmentation of learned movement representations. For this purpose, a learning cycle is presented which in addition to the imitation learning paradigm incorporates the concept of *Structural Bootstrapping* that has been introduced in the Xperience (Xperience Project, 2011) to address how generative mechanisms which rely on prior knowledge and sensorimotor experience can be implemented in robotic systems and employed to speed up learning. Structural Bootstrapping – an idea taken from child language acquisition research – is a method which provides an explanation of how the language acquisition process in infants is initiated. Hence, in a robotic context, Structural Bootstrapping can be seen as a method of building generative models, leveraging existing experience to predict unexplored action effects and to focus the hypothesis space for learning novel concepts. This developmental approach enables rapid generalization and acquisition of new knowledge about objects, actions and their effects from little additional training data. In the following, focusing on the scenario of table wiping, an example for Structural Bootstrapping is provided in order to demonstrate the validity of the approach on the sensorimotor motor level.

### C.1. Learning of Affordances from Exploration and Physical Interaction

To enhance to efficiency robot manipulation in unstructured and dynamic environments, robots have to be endowed with the cognitive capabilities which enable the autonomous acquisition of knowledge by processing complex sensor information and the application of this knowledge to rapidly explore unknown scenes, objects, and actions. The experience gained by actively exploring and interacting with the environment, objects, and other agents and by observing the effect of actions is characterized by the specific embodiment. Therefore, representations and models emerging from this experience are better adapted to the robot's morphology and more suitable to capture the sensorimotor contingencies than those generated by traditional disembodied methods. The continuous grounding of internal models and representations through exploration provides a suitable basis for the reasoning of a wiping action. In the context of the table wiping task, a learning cycle based on Structural Bootstrapping is instantiated in order to allow the generation of models which describe the relation between object properties and action parameters learned from experience and is used to make predication using internal simulation.

Several approaches in the literature deal with the problem of exploration-based learning and generative model construction. In (Montesano et al., 2007), an affordance learning framework is introduced which models dependencies between action and object features in the form of a Bayesian Network. Using a set of manipulation actions (grasp, tap, touch) and based on perceived object features the expected effect of an action to be performed could be estimated. In (Kroemer et al., 2012), an interactive learning scheme is introduced which allows the identification of object grasp affordances. Grasp primitives represented in the form of DMPs learned from human grasp demonstrations and grounded based the observed effect (grasp

successful or not). Towards structural bootstrapping, in (Detry et al., 2011), an approach is presented for the learning of object grasp affordances through exploration. These affordances are represented by grasp densities which are determined based on the visual features (3D edges) of the object to be grasped. The object grasp affordances are grounded and the grasp densities are refined based on exploration and observation of grasping actions performed by the robot. In (Ugur et al., 2012), an approach is introduced which enables a robot to learn a grasping behavior based on initial reflex-like motor primitives. The execution of these primitives at different speeds and the observation of the tactile feedback when touching an object leads to the generation of further behavior primitives. To link the resulting behavior to different intrinsic and extrinsic object properties, the primitives are executed and the observed effects are categorized using the Support Vector Machine. For the scenario of object-pushing, in (Hermans et al., 2013), a method is proposed which enables a robot to learn goal-directed push-locations on multiple objects. Using the Support Vector Regression method a model is learned from explorative pushing which allow the prediction of the effect of certain pushing action considering the current object shape and pose.

## C.2. The Learning Cycle

In order to enable a robotic system to learn and refine sensorimotor knowledge within a developmental process, a learning cycle has to be formalized which incorporates perceptual and motor skills. As suggested in (Kolb, 1984), the presented learning cycle consists of four stages. The initial stage of this cycle is the exploration stage. Given generalized representations of objects and actions, the robot explores the scene in order to obtain instantiations of both, object and action. The resulting action and object representation  $A_1$  and  $P_1$  form the basis of an experiment which is conducted in the subsequent stage to create data from which concrete experience can be generated. The robot applies the action  $A_1$  and observes its effect  $E_1$  on object, environment, and on the robot itself. In the third stage, based on the data  $D = (P_1, A_1, E_1)$  experience is created by grounding and adapting the representations. In the modeling stage, knowledge in the form of internal models  $f_E$  and  $f_A$  is extracted from experiential data.

In subsequent iterations  $i$  with  $i > 1$ , the grounding is transferred to novel perceived object representation  $P_i$ . Using  $f_A$  and  $f_E$  the parameters for action  $\hat{A}_i$  and the expected effect  $\hat{E}_i$  for  $(P_i, \hat{A}_i)$  can be predicted.  $\hat{A}_i$  can be used to constrain and control the exploration of the action parameter space within the repeated experiment and with  $\hat{E}_i$  less, however, more relevant additional training data can be created which has to be considered for the re-grounding the representations and revision of the internal models. Hence, this learning cycle allows the continuous acquisition, validation, and refinement of internal knowledge in long term association through exploration and predictive reasoning.

### C.2.1. Instantiation of the Learning Cycle for Wiping

Based on the concept described in Section C.2, as presented in (Do et al., 2014), a behavior is implemented which enables a robot to efficiently learn wiping movements with different objects. The resulting learning cycle is depicted in Figure C.1. To accelerate the learning process, observations of human wiping demonstrations trigger the bootstrapping process and provide data based on which a coarse representation of the wiping action can be inferred. As described Appendix B, the wiping action is represented in the generalized form of a periodic DMP. In the initial iteration, the robot is focused on the adaptation of this representation to environmental circumstances, namely the surface to be wiped. This step corresponds to

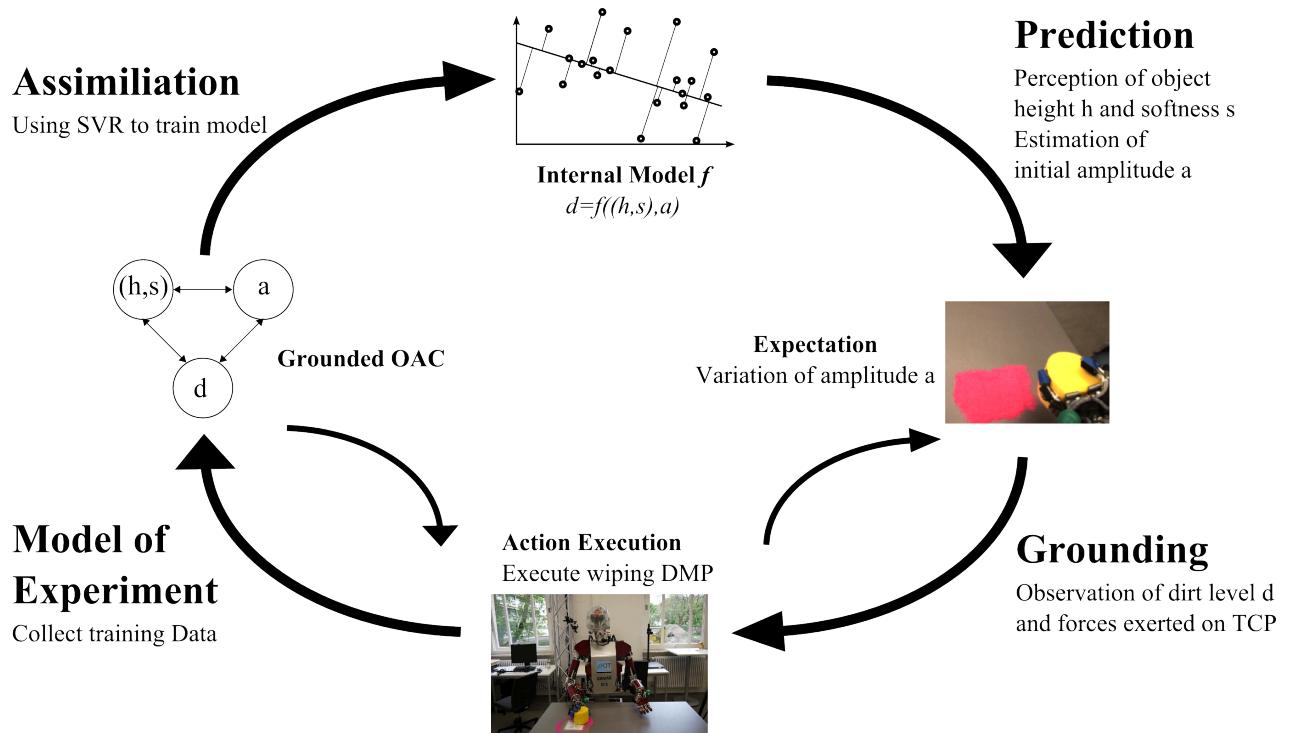


Figure C.1.: Instantiated learning cycle for the learning of wiping.

the grounding of the action representation. In subsequent iterations, the robot attempts to establish the link between a object, action, and effect.

### C.2.2. Exploration of Object Features

Each iteration starts with an object exploration phase through manipulation and perception in order to obtain discriminative object features. Thus, in the context, the size and softness of the object have been determined to be relevant. To determine the softness of an object the robot uses its ability to control the grasping force of the pneumatically actuated hand with a model-based force position control (see (Bierbaum et al., 2009b)). Initially the object is grasped between the fingertips with a low grasping force at the top and the bottom side. The distance between the fingertips of the index finger, middle finger, and thumb is measured using the joint encoders and the forward kinematics. This initial distance value denotes the height of the object  $h$ . Subsequently, the grasping force is increased which results in a deformation of the object. Once a desired grasping force is attained, the distance between the fingertips is measured again. The ratio between both distance measurements is used as a measure for the softness  $s$  of the object.

### C.2.3. Exploration of Action Parameters

Based on the object representation  $(s, h) \in \mathbb{R}^2$ , an experiment is conducted in order to find out the action parameters which optimize the effect of an action given the explored object. To generate differently scaled wiping movements, the amplitude parameter incorporated in the learned periodic DMP representation can be varied. Especially, regarding the movement of the endeffector directed towards the table, the amplitude has to be scaled according to the specific softness parameter. The search for the optimal amplitude parameter  $a$  entails considerable effort since it involves the variation of  $a$ , the subsequent parameterization of the



Figure C.2.: Left: Robot view from the scene. Center: Segmented view of the scene in the beginning of the wiping execution. Right: Segmented view on a "clean" table.

wiping primitive, and the reproduction of a wiping action. An important indicator whether a wiping movement is parameterized and executed adequately is given by the forces acting in the robot. Therefore, starting from an initial estimate  $a_0$ , the amplitude is varied according following rules:

$$a(t) = \begin{cases} b^- a(t-1) & , f_{z_w}(t) - f_0 > \rho, \dot{z}_w < 0 \\ b^+ a(t-1) & , f_{z_w}(t) - f_0 > \rho, \dot{z}_w > 0 \\ b^+ a(t-1) & , f_{z_w}(t) - f_0 < -\rho, \dot{z}_w < 0, \\ b^- a(t-1) & , f_{z_w}(t) - f_0 < -\rho, \dot{z}_w > 0 \\ a(t-1) & \text{else} \end{cases} \quad (\text{C.1})$$

where  $0 < b^- \leq 1$  and  $b^+ = 2 - b^-$  denotes a scalar factor which decreases respectively increases the amplitude according the current movement direction and exerted forces. To accommodate potential noise contaminating the force torque sensor readings, instead of fixating the desired surface pressure on  $f_0$ ,  $\rho$  is introduced into the amplitude update rule to define a range of force values  $[f_0 - \rho, f_0 + \rho]$  in which the forces acting on the endeffector are considered to be optimal.  $a(t-1)$  represents the amplitude estimate made in the previous time step. For each iteration  $i$ , the overall amplitude factor  $a_i$  is calculated by  $a_i = \frac{1}{T_c^+ - T_c^-} \sum_{t=T_c^-}^{T_c^+} a(t)$ .

A further cue which allows the evaluation of a wiping movement is the dirt level which describes the ratio between the amount of remaining dirt enclosed by an area to be wiped and the entire wiping area size. While the action parameter space is explored based on the force cue, the progress of the experiment is controlled by the dirt level. Thus, the goal of the experiment is to wipe until the dirt level does not change. For dirt levels  $d_i, d_{i-1} \in \mathbb{R}$  determined in iteration  $i$  and  $i-1$ , the goal can be formalized as follows:

$$d_i - d_{i-1} \leq d_\epsilon \quad (\text{C.2})$$

where  $d_\epsilon$  denote a threshold at which the dirt level change can be disregarded.

#### C.2.4. Exploration of Effect

As mentioned before, the effect of a wiping action is described by the dirt level  $d$  within area  $O$  to be wiped. For the sake of simplicity, it is assumed that dirt features a predefined color. Therefore, to determine the size and position of  $O$ , using the stereo camera setup the robot explores the table and performs a color segmentation in order to localize the largest blob. A bounding box  $B_i$  around that blob provides the image coordinates of  $O$ . Transformed into the world coordinate system, one obtains  $B_w$  which provide the global coordinates of  $O$ . In order to determine the current dirt level at any time  $t$  during the execution of the wiping experiment,

$B_w$  is transformed back onto image coordinates  $B_i^t$ . Hence, based on  $B_i^t$  the current dirt level  $d_i(t)$  is calculated according to following equation:

$$d_i(t) = \sum_{i=y_{min}}^{y_{max}} \sum_{j=x_{min}}^{x_{max}} \frac{k(i, j)}{(x_{max} - x_{min})(y_{max} - y_{min})}. \quad (C.3)$$

Since the hand might occlude a considerable area of the surface, a reliable assessment of the dirt level cannot be guaranteed at any time during the execution of a wiping movement. Hence, to control the experiment, the current dirt level at  $t_c$  is set to  $d(t_c) := d_{max,i}$  which is defined as follows:

$$d_{max,i} = \max \{d_i(t)\}_{T_{s_i} < t < t_c} \quad (C.4)$$

with  $i$  denoting the index of the current period.

### C.2.5. Learning of Internal Models

To enhance the adaptation of a wiping primitive to novel objects, based on sensorimotor experience gathered in previous iterations, internal knowledge structures are derived in the form of models which are used for the prediction of the expected wiping effect for a specific object-action combination. These models are trained from data which has been collected during the conducted wiping experiment. The data can be described as follows:

$$(P, A, E) = ((s, h), a, d). \quad (C.5)$$

To generate an internal model representing the complex relationships between perception, action, and effect, computational methods have to be applied which are suitable to identify structures from nonlinear data of arbitrary dimensionality without any prior knowledge. In this thesis, the Support Vector Regression, a supervised learning technique which is described in (Smola, 2011), is applied to approximate such a model, since it allows to capture complex relationships between the training data points. Furthermore, a sparse model can be obtained by applying the Support Vector method which facilitates the processing of large datasets and enhances the prediction and simulation using the internal model. Based on our experimental data collection  $\{(P_n, A_n, E_n)\}_{i=1, \dots, N}$ , for the training of  $f_E$ , a dataset  $D$  with  $N$  input/output pairs is formed as follows:

$$D = \{(x_n, y_n)\}_{n=1, \dots, N}, \quad x_i = (P_i, A_i), \quad y_i = (E_i). \quad (C.6)$$

The internal model is described by  $f_E : x \rightarrow y$ . Finding a nonlinear mapping appropriate function  $f_E$  solves the learning problem and leads to desired model enabling the mapping of an arbitrary input pair  $(P, A)$  on expected effect  $\hat{E}$ . Usually, the search for  $f_E$  is performed by determining an approximation  $\hat{f}_E$  which minimizes the risk functional:

$$R_{emp} [\hat{f}_E] := \frac{1}{N} \sum_{n=1}^N d(\hat{f}_E(x_n), y_n) \quad (C.7)$$

with  $d(f_E(x), y)$  being a distance function to define the relation between the model's output  $\hat{f}_E(x)$  and the correct output  $y$ . Using the Support Vector method, the nonlinear regression problem incorporated in Eq. C.7 is transformed into linear problem by introducing a nonlinear mapping  $\theta : \mathbb{R} \rightarrow \mathbb{R}^{N_h}$  which projects the original dataset  $D$  into a feature space of higher dimensionality. Hence, the SVR consists of finding a hyperplane  $(w, b)$  which satisfies:

$$g(x, w) = \sum_{j=1}^{N_h} w_j \theta_j(x) + b. \quad (C.8)$$

To determine a linear model which captures most training samples within an  $\varepsilon$ -margin, an  $\varepsilon$ -loss-insensitive function is defined as follows:

$$L_\varepsilon(g(x, w), y) = \begin{cases} 0 & \text{if } |g(x, w) - y| \leq \varepsilon \\ |g(x, w) - y| - \varepsilon & \text{else} \end{cases} \quad (\text{C.9})$$

is introduced into the risk functional. Hence, our goal is to find a function  $f_E$  whose distance to any given data point does not exceed  $\varepsilon$  while being as flat as possible. This optimization problem can be described:

$$\text{minimize } \tau(w) = \frac{1}{2} \|w\|^2 + C \sum (\zeta + \zeta^*) \quad (\text{C.10})$$

$$\text{subject to } y_i - (g(x_i, w) - b) \leq \varepsilon \quad (\text{C.11})$$

$$\text{subject to } (g(x_i, w) + b - y_i) \leq \varepsilon, \quad (\text{C.12})$$

where  $\zeta$  are slack variables which are introduced to the problem in order to relax the constraints and to add a soft margin to the hyperplane and, thus, to tolerate a small error.  $C$  is a constant which controls the trade-off between the flatness of  $f_E$  and the tolerated deviations larger than  $\varepsilon$ . Since  $\theta$  is unknown according (Smola, 2011) a suitable kernel function such as the Radial Basis Function:

$$k(x, x_i) = \exp(-\gamma \|x - x_i\|) \quad (\text{C.13})$$

can be used to instead in order to project the data into high-dimensional space. The main parameters controlling the performance of the SVR method are  $C$  and the kernel parameter  $\gamma$ .

### C.3. Experiments

As depicted in Figure C.3, the implemented learning behavior has been evaluated on our humanoid platform ARMAR-IIIb. Each iteration of the learning cycle starts with the robot localizing the dirty area  $\mathcal{O}$ . To facilitate the environmental perception, color of the dirt has been specified (pink sand). The corresponding bounding box  $B_i$  is used to specify the target configuration of the DMP. In the following step, the robot determines the object softness and height by grasping the object of interest at the bottom and top side of the object. The object exploration process is assisted by human operator since for wiping the object has to be reoriented in the robot's hand, so that the object is grasped from the side enabling the bottom to touch the table. Given the internal models, predictions are made for the amplitude and the expected effect. Subsequently, the robot performs a wiping movement with  $a$  and compares the observed effect with the expected effect. If the observations does not coincide with the expectation, a parameter exploration procedures as described in Section C.2.3 amplitude is initiated in order to create further data for the grounding of the internal models. The grounding of an internal model is done by updating the data set and retraining the entire model.

The wiping experiments have been conducted on set of twelve objects which includes instances designed for wiping (sponges, towel, toilet paper) and other household items (box, bottle, ball, can) that are less suitable. To prevent damage to the robot restrict, objects have been selected whose height and weight are within a predefined range. Based on experimental data originating from wiping experiments with this object set, internal models  $f_E$  and  $f_A$  are generated using the SVR method. Based on the implementation incorporated in the LIBSVM library (see (Chang and Lin, 2011)), the SVR method used in this thesis has



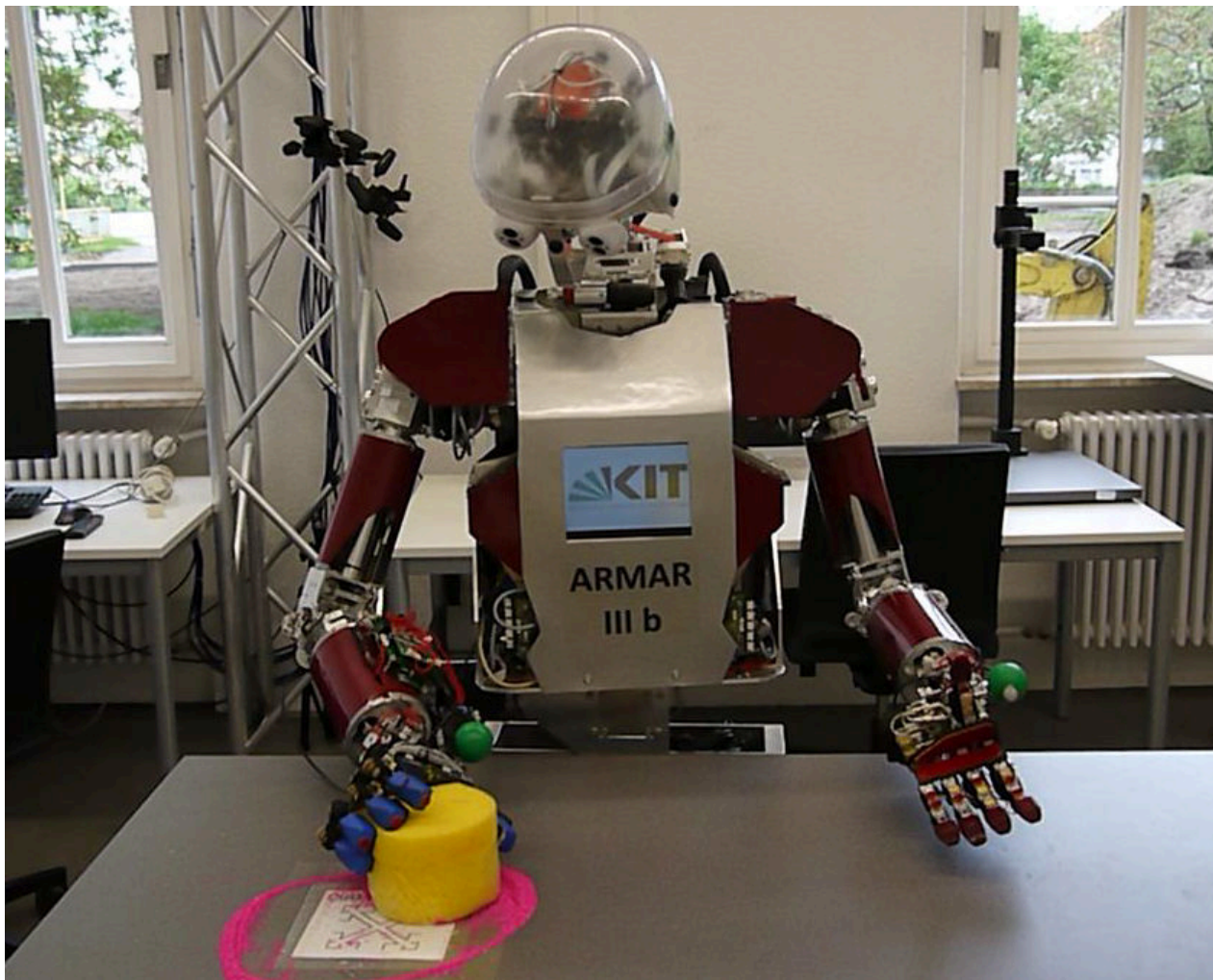


Figure C.3.: The humanoid platform ARMAR-IIIb wiping the table with a sponge.

been implemented in the ClaRe framework. The relevant parameters for the training of  $f_A$  have been determined to be  $C = 50$  and  $\gamma = 0.5$  whereas  $f_E$  has been trained with  $C = 10$  and  $\gamma = 0.33$ . The data and predictions of the amplitudes and the expected dirt levels are listed in Table C.1. It is interesting to see that for soft objects the amplitude could be reliably re-estimated. The main reason for the variation of the amplitudes for harder objects lies in the increased sensitivity towards forces exerted on the object respectively the end-effector. A slight difference of the object pose in hand can produce very different results. Regarding the prediction of the expected dirt level, good estimations could be made for cubic objects. For spherical and cylindric objects, less useful predictions have been inferred.

Given a percept of a specific object, the corresponding amplitude estimate can be used to considerably reduce the adaptation effort of a wiping movement. The plots depicted in Figure C.4 indicate that with increasing knowledge leading to more accurate estimations of the action parameter the execution of an action converges faster towards the desired behavior. With regard to the forces exerted on the end-effector, a force trajectory is desired which oscillates around the predefined force threshold of  $f_0 = 15$  whereas regarding the dirt level it is desired to minimize the dirt level as fast as possible. The learning phase denotes the initial phase where the movement primitive is adapted to the environment. In the adaptation phase, based on a default value of  $a_0 = 1$  the amplitude is varied in order to attain the desired effect. In the execution phase, the task is performed using the estimated amplitude parameter and without any adaptation.

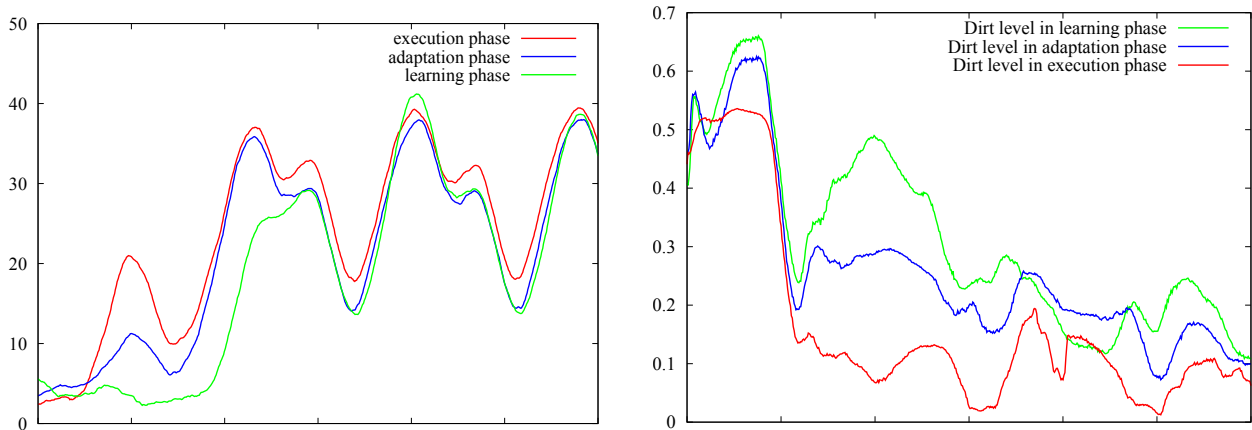


Figure C.4.: Left: Trajectories of forces exerted on the end effector in various phases of the wiping learning cycle. Right: Dirt level evolution in various phases of the wiping learning cycle.

### C.4. Conclusion

An approach for the implementation of a cognitive learning behavior enabling robots to create individual knowledge structures based on experience gained through physical exploration, interaction, and observation has been proposed. The behavior manifests in the form a learning cycle which incorporates perceptual and motor skills in order to continuously acquire data based on which internal models are generated and grounded. For the scenario of table wiping, it has been shown that with these internal models wiping primitives can be efficiently learned and adapted to different task and object-specific constraints.

Object	$h$	$s$	$\hat{a}$	$a$	$\hat{d}$	$d$
sponge (s)	79	0.0343	1.0	1.0	0.162	0.117
sponge (m)	91	0.0384	0.957	0.948	0.129	0.132
sponge (l)	102	0.0358	1.13	1.139	0.139	0.09
styrofoam cube (s)	87	0.00474	0.701	0.696	0.270	0.258
styrofoam cube (l)	91	0.00774	1.0	1.0	0.13	0.177
rolled towel	89	0.0213	1.215	1.057	0.229	0.142
styrofoam ball	100	0.00639	1.497	1.496	0.384	0.422
cardboard box	91	0.0171	1.072	1.072	0.240	0.178
plastic bottle	87	0.0263	1.453	1.453	0.310	0.568
metal can	86	0.00843	1.04	1.366	0.352	0.529
toilet paper	101	0.0232	0.887	0.887	0.162	0.128
foam	91	0.041	0.999	1.0	0.13	0.177

Table C.1.: Object properties and the corresponding action and effect parameter.  $h$  denotes the object height in mm and  $s$  the softness of an object.  $\hat{a}$  and  $a$  represent the estimated and the actual amplitude of an adapted wiping movement.  $\hat{d}$  and  $d$  stand for the expected and actual dirt level which indicates the effect of wiping.

## List of Figures

1.1	Examples for human and robotic grasping . . . . .	2
1.2	Overview of the grasp learning cycle . . . . .	3
2.1	Kamakura grasp taxonomy . . . . .	8
2.2	Cutkosky grasp taxonomy . . . . .	9
2.3	Hoff-Arbib model . . . . .	10
2.4	Grasp models based on individual digits . . . . .	11
2.5	Principal Components of the grasp configuration space . . . . .	12
3.1	Various robotic hand systems . . . . .	13
3.2	Grasp mapping (see (Tegin et al., 2009)) and grasp synthesis approach (see (Li and Pollard, 2005)) . . . . .	14
3.3	Illustration of Eigengrasps (see (Ciocarlie and Allen, 2009)) . . . . .	15
3.4	Illustration of the Contact Web (see (Kang and Ikeuchi, 1997)) . . . . .	16
3.5	Manipulation manifolds for cap turning movement (see (Steffen et al., 2008)) .	17
3.6	Grasp representation based on coupled dynamical systems approach (see (Shukla and Billard, 2012)) . . . . .	18
3.7	Illustration of DMP-based approaches (see (Kroemer et al., 2010),(Stulp et al., 2011)) . . . . .	19
3.8	Grasp representation based on clusters in latent space (see (Palm and Iliev, 2006)) . . . . .	20
3.9	GMM-based approach (see (Romero et al., 2010)) . . . . .	21
4.1	2D vector field describing an attractor landscape . . . . .	24
4.2	Backprojection error for grasp examples represented in 2D latent space . . . .	25
4.3	Finger movement synergies in the task space . . . . .	26
4.4	Plots depicting the behavior of linear mass spring damper system with varying damping ratio and spring constant . . . . .	27
4.5	Illustration of a linear spring and a multi-body system . . . . .	28
4.6	Illustration of the Virtual Spring Hypothesis . . . . .	29
4.7	The VSG Representation for a three-finger tripod grasp . . . . .	30
4.8	Visualization of the Virtual Contact Strip . . . . .	33
4.9	Illustration of the start, preshape, and enclose configurations of a spherical precision grasp using the VSG representation . . . . .	35
5.1	Illustration of the Vicon marker configuration . . . . .	40
5.2	Sample images from the markerless upper body tracking procedure . . . . .	44
5.3	Sample images from the hand tracking procedure . . . . .	45
5.4	Processing line of the proposed fingertip tracking approach . . . . .	46
5.5	Feature images from the fingertip tracking procedure . . . . .	47
5.6	Sample images from the fingertip tracking procedure . . . . .	48
5.7	Illustration of particles in the fingertip tracking procedure . . . . .	49
5.8	Illustration of the fingertip estimates correction step . . . . .	50
5.9	Illustrations of the marker transformation . . . . .	52

6.1	Flowchart of the proposed estimation scheme . . . . .	58
6.2	Illustration of regression examples for least squares and TLS solver . . . . .	60
6.3	Visualization of the error for the optimization of spring constants . . . . .	61
7.1	Master Motor Map . . . . .	71
7.2	Illustration of the proposed hand IK solution . . . . .	75
7.3	System architecture . . . . .	77
8.1	Images of the experimental platforms: The humanoid robot ARMAR-IIIb and Motoman manipulator with a Gifu hand . . . . .	80
8.2	Sample images of the fingertip tracking results . . . . .	81
8.3	Plot depicting the pixel error for the fingertip tracking procedure . . . . .	82
8.4	Comparison between demonstrated and reproduced fingertip trajectories using the VSG representation . . . . .	84
8.5	Convergence behavior of the proposed estimation scheme for markerless and marker-based motion data . . . . .	85
8.6	Trajectories of demonstrated and reproduced prehensile fingertip movements using VSG representations which encode volar grasp types . . . . .	88
8.7	Trajectories of demonstrated and reproduced prehensile fingertip movements using VSG representations which encode non-volar grasp types . . . . .	89
8.8	Adaptation and reproduction of a pick and place task using DMPs . . . . .	91
8.9	Reproduced DMP movements for different actions . . . . .	92
8.10	Sample images of the shell game reproduction using DMPs . . . . .	93
8.11	Temporal evolution of the grip aperture movement for different object scales . . . . .	94
8.12	Plots comparing reproduced grip aperture movements using the VSG, the DMP, and the GMM-based approaches . . . . .	96
8.13	Illustration of mapped arm configurations . . . . .	99
8.13	Sample images from reproduction of various grasp primitives learned from marker-based motion . . . . .	101
8.14	Illustration of the evolution of the Virtual Contact Strip . . . . .	102
8.15	Illustrations of non-volar grasp types learned from human observation in an online manner . . . . .	102
8.16	Reproduced grasp preshapes and different scales . . . . .	103
8.17	Final grasp postures of reproduced grasp primitives adapted to different object scales . . . . .	104
A.1	Software components of the grasp learning framework . . . . .	109
A.2	Illustration of Vicon marker trajectories and their transformation into the robot hand coordinate system . . . . .	111
A.3	Snapshots of the graphical user interface of the grasp learning framework . . . . .	113
B.1	Plots of different wiping styles . . . . .	118
B.2	Adaptation of periodic transient-encoding DMP to different start and goal positions . . . . .	119
B.3	Reproduction of wiping primitives on the humanoid robot ARMAR-IIIb . . . . .	121
C.1	Instantiated learning cycle for the learning of wiping . . . . .	125
C.2	Robot views from the scene . . . . .	126
C.3	The humanoid platform ARMAR-IIIb wiping the table with a sponge . . . . .	129
C.4	Plots depicting the forces acting on robot endeffector and the dirt level . . . . .	130

## List of Tables

8.1	Comparison between the proposed parameter estimation scheme and gradient-based optimization algorithm . . . . .	85
8.2	Spring constant estimates representing various grasp types . . . . .	87
8.3	Spring constant estimates for similar grasp examples . . . . .	90
C.1	Object properties and the corresponding action and effect parameter . . . . .	130



## Bibliography

- Amor, H. B., Kroemer, O., Hillenbrand, U., Neumann, G., and Peters, J. (2012). Generalization of human grasping for multi-fingered robot hands. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2043–2050.
- Andrieu, C., Doucet, A., and Tadic, V. B. (2005). On-line parameter estimation in general state-space models. In *IEEE Conference on Decision and Control and European Control Conference*, pages 332–337.
- Arbib, M. A., Iberall, T., and Lyons, D. (1985). *Coordinated control programs for control of the hands*, volume 10, pages 111–129. Springer, Berlin.
- Argyros, A. A. and Lourakis, M. (2006). Vision-based Interpretation of Hand Gestures for Remote Control of a Computer Mouse. In *Proceedings of HCI Workshop*, pages 40–51.
- Arimoto, S., Sekimoto, M., Hashiguchi, H., and Ozawa, R. (2005). Natural resolution of ill-posedness of inverse kinematics for redundant robots: a challenge to Bernstein’s degrees-of-freedom problem. *Advanced Robotics*, pages 401–434.
- Asfour, T., Azad, P., Gyarfas, F., and Dillmann, R. (2008). Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots. *International Journal of Humanoid Robotics*, 5(2):183–202.
- Asfour, T., Gyarfas, F., Azad, P., and Dillmann, R. (2006a). Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 40–47, Genova, Italy.
- Asfour, T., Regenstein, K., Azad, P., Schröder, J., Vahrenkamp, N., and Dillmann, R. (2006b). ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 169–175, Genova, Italy.
- Azad, P. (2009). *Visual Perception for Manipulation and Imitation in Humanoid Robots*. Springer.
- Azad, P. (2011). Integrating Vision Toolkit (IVT). Available online at <http://ivt.sourceforge.net>.
- Azad, P., Asfour, T., and Dillmann, R. (2007). Stereo-based 6D Object Localization for Grasping with Humanoid Robot Systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 919–924, San Diego, USA.
- Azad, P., Asfour, T., and Dillmann, R. (2008). Robust Real-time Stereo-based Markerless Human Motion Capture. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 700–707, Daejeon, Korea.
- Azad, P., Asfour, T., and Dillmann, R. (2009). Combining Harris Interest Points and the SIFT Descriptor for Fast Scale-Invariant Object Recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4275–4280, St. Louis, USA.

- Bakker, P. and Kuniyoshi, Y. (1996). Robot see, robot do: An overview of robot imitation. In *AISB'96 Workshop of Learning in Robots and Animals*, pages 3–11.
- Becedas, J., Mamani, G., Feliu, V., and Sira-Ramirez, H. (2009). *Estimation of Mass-Spring-Damper Systems*, pages 411–422. Advances in Computational Algorithms and Data Analysis. Springer, Netherlands.
- Berenson, D., Diankov, R., Nishiwaki, K., Kagami, S., and Kuffner, J. (2007). Grasp planning in complex scenes. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 42–48.
- Bicchi, A., Gabbicini, M., and Santello, M. (2011). Modeling Natural and Artificial Hands with Synergies. *Philosophical Transactions of the Royal Society B*, 366:3153–3161.
- Bierbaum, A., Asfour, T., and Dillmann, R. (2009a). Dynamic Potential Fields for Dexterous Tactile Exploration. In *Human Centered Robot Systems*, pages 23–31. Springer.
- Bierbaum, A., Schill, J., Asfour, T., and Dillmann, R. (2009b). Force Position Control for a Pneumatic Anthropomorphic Hand. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 21–27, Paris, France.
- Blake, A. and Isard, M. (2000). *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer.
- Borst, C., Fischer, M., and Hirzinger, G. (2004). Grasp Planning: How to Choose a Suitable Task Wrench Space. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 319–325, New Orleans, USA.
- Bregler, C. and Malik, J. (1998). Tracking People with Twists and Exponential Maps. In *International Conference on Computer Vision and Pattern Recognition*, pages 8–15.
- Bretzner, L., Laptev, I., and Lindeberg, T. (2002). Hand Gesture Recognition using Multi-Scale Colour Features, Hierarchical Models and Particle Filtering. In *Proc. Int. Conf. Aut. Face and Gesture Recognition*, pages 423–428.
- Brown, C. Y. and Asada, H. H. (2007). Inter-finger coordination and postural synergies in robot hands via mechanical implementation of principal components analysis. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2877–2882.
- Buchholz, B., Armstrong, T. J., and Goldstein, S. A. (1992). Anthropometric data for describing the kinematics of the human hand. *Ergonomics*, 35(3):261–273.
- Bullock, I. M. and Dollar, A. M. (2011). Classifying Human Manipulation Behavior. In *IEEE International Conference on Rehabilitation Robotics*, Zurich, Switzerland.
- Burns, A. M. and Wanderley, M. M. (2006). Visual Methods for the Retrieval of Guitarist Fingering. In *Proc. International Conference on New Interfaces for Musical Expression*, pages 196–199, Paris, France.
- Calinon, S., Guenter, F., and Billard, A. (2007). On Learning, Representing and Generalizing a Task in a Humanoid Robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, 37(2):286–298.



- Castiello, U. (2005). The neuroscience of grasping. *Nature Reviews Neuroscience*, 6(9):726–736.
- Catalano, M., Giorgio, G., Serio, A., Farnioli, E., Piazza, C., and Bicchi, A. (2012). Adaptive synergies for a humanoid robot hand. In *IEEE-RAS international conference on humanoid robots*.
- Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Chen, W. and Xiong, C. (2013). A Principle of Mechanical Implementing the Kinematic Synergy for Designing Anthropomorphic Hand. *Lecture Notes in Computer Science*, 8102:339–350.
- Ciocarlie, M. T. and Allen, P. K. (2009). Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research*, 28:851–867.
- Cutkosky, M. R. (1989). On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics*, 5:269–279.
- Demirdjian, D., Ko, T., and Darrell, T. (2003). Constraining Human Body Tracking. In *International Conference on Computer Vision*, pages 1071–1078.
- Detry, R., Kraft, D., Kroemer, O., Bodenhausen, L., Peters, J., Krüger, N., and Piater, J. (2011). Affordance Prediction via Learned Object Attributes. *Journal of Behavioral Robotics*, 2(1):1–17.
- Deutscher, J., Blake, A., and Reid, I. (2000). Articulated Body Motion Capture by Annealed Particle Filtering. In *International Conference on Computer Vision and Pattern Recognition*, pages 2126–2133.
- Do, M., Asfour, T., and Dillmann, R. (2011a). Particle Filter-Based Fingertip Tracking with Circular Hough Transform Features. In *IAPR Machine Vision Applications*.
- Do, M., Asfour, T., and Dillmann, R. (2011b). Towards a Unifying Grasp Representation for Imitation Learning on Humanoid Robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 482–488.
- Do, M., Azad, P., Asfour, T., and Dillmann, R. (2008). Imitation of Human Motion on a Humanoid Robot using Nonlinear Optimization. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 545–552, Daejeon, Korea.
- Do, M., Romero, J., Kjellström, H., Azad, P., Asfour, T., Kragic, D., and Dillmann, R. (2009). Grasp Recognition and Mapping on Humanoid Robots. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Paris, France.
- Do, M., Schill, J., Ernesti, J., and Asfour, T. (2014). Learn to Wipe: A Case Study of Structural Bootstrapping from Sensorimotor Experience. In *IEEE International Conference on Robotics and Automation (ICRA)*.
- Ekvall, S. and Kragic, D. (2004). Interactive grasp learning based on human demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3519–3524.

- Elder, J. H. and Zucker, S. W. (1996). Scale Space Localization, Blur, and Contour-Based Image Coding. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*.
- Erickson, D., Weber, M., and Sharf, I. (2003). Contact stiffness and damping estimation for robotic systems. *The International Journal of Robotics Research*, 22(1):41–57.
- Ernesti, J., Righetti, L., Do, M., Asfour, T., and Schaal, S. (2012). Encoding of Periodic and their Transient Motions by a Single Dynamic Movement Primitive. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 57–64, Osaka, Japan.
- Feix, T., Schmiedmayer, H.-B., Romero, J., and Kragic, D. (2009). A comprehensive grasp taxonomy. In *Robotics, Sciences and Conference: Workshop on Understanding The Human Hand for Advancing Robotic Manipulation*.
- Fischer, A., Do, M., Stein, T., Asfour, T., Dillmann, R., and Schwameder, H. (2011). Recognition of Individual Kinematic Patterns during Walking and Running - A Comparison of Artificial Neural Networks and Support Vector Machines. *International Journal of Computer Science in Sports*, 10(1).
- Flacco, F. and Luca, A. D. (2011). Residual-based stiffness estimation in robots with flexible transmissions. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5541–5547.
- Gaiser, I., Schulz, S., Kargov, A., Klosek, H., Bierbaum, A., Pylatiuk, C., Oberle, R., Werner, T., Asfour, T., Bretthauer, G., and Dillmann, R. (2008). A new anthropomorphic robotic hand. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 418–422, Daejeon, Korea.
- Gams, A., Do, M., Ude, A., Asfour, T., and Dillmann, R. (2010). On-Line Periodic Movement and Force-Profile Learning for Adaptation to New Surfaces. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Nashville, USA.
- Gärtner, S., Do, M., Simonidis, C., Asfour, T., Seemann, W., and Dillmann, R. (2010). Generation of Human-like Motion for Humanoid Robots Based on Marker-based Motion Capture Data. In *Proceedings for the joint conference of ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*.
- Gautier, M. and Pognet, P. (2001). Extended Kalman filtering and weighted least squares dynamic identification of robot. *Control Engineering Practice*, 9(12):1361–1372.
- Golub, G. H. and Loan, C. F. V. (1983). *Matrix Computations*. John Hopkins University Press, Baltimore, USA.
- Haggard, P. and Wing, A. (1997). On the hand transport component of prehensile movements. *Journal of Motor Behavior*, 29:282–287.
- Hermans, T., Li, F., Rehg, J. M., and Bobick, A. (2013). Learning Stable Pushing Locations. In *IEEE International Conference on Developmental Learning and Epigenetic Robotics (ICDL-EPIROB)*, Osaka, Japan.
- Hersch, M., Guenter, F., Calinon, S., and Billard, A. (2008). Dynamical System Modulation for Robot Learning via Kinesthetic Demonstrations. *IEEE Transactions on Robotics*, 24:1463–1467.

- Hoff, B. and Arbib, M. A. (1993). Models of Trajectory Formation and Temporal Interaction of Reach and Grasp. *Journal of Motor Behavior*, 25:175–192.
- Hsiao, K. J., Chen, T. W., and Chien, S. Y. (2008). Fast fingertip positioning by combining particle filtering with particle random diffusion. In *Proc. International Conference on Multimedia and Expo*, pages 977–980.
- Iberall, T. (1986). The representation of objects for grasping. In *Eighth Annual Conference of the Cognitive Science Society*, pages 547–561, Amherst, USA.
- Iberall, T. (1997). Human prehension and dexterous robot hands. *International Journal of Robotics Research*, 16:285–299.
- Ijspeert, A. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–653.
- Ijspeert, A., Nakanishi, J., Pastor, P., Hoffmann, H., and Schaal, S. (2013). Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. (25):328–373.
- Ijspeert, A. J., Nakanishi, J., and Schaal, S. (2002). Learning Attractor Landscapes for Learning Motor Primitives. In *Conference on Neural Information Processing Systems*, pages 1523–1530.
- Inamura, T., Toshima, I., and Nakamura, Y. (2003). *Experimental Robotics VIII, Vol. 5*, chapter Motion Elements for Bidirectional Computation of Motion Recognition and Generation, pages 372–381. Springer, Berlin, Germany.
- Isard, M. and Blake, A. (1998). CONDENSATION – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28.
- Jacobsen, S. C., Iversen, E. K., Knutti, D., Johnson, R., and Biggers, K. (1986). Design of the Utah/M.I.T. Dextrous Hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1520–1532.
- Jäkel, R. (2013). *Learning of Generalized Manipulation Strategies in Service Robotics*. PhD thesis, Karlsruhe Institute of Technology (KIT).
- Jäkel, R., Schmidt-Rohr, S. R., Xue, Z., Lösch, M., and Dillmann, R. (2010). Learning of probabilistic grasping strategies using Programming by Demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*.
- Jeannerod, M. (1981). Intersegmental coordination during reaching at natural visual objects. *Attention and Performance*, 9:153–168.
- Johnson, S. G. (2013). The NLOpt nonlinear-optimization package. Available online at <http://ab-initio.mit.edu/nlopt>.
- Kamakura, N. (1980). Patterns of static prehension in normal hands. *The American journal of occupational therapy. : official publication of the American Occupational Therapy Association*, 34:437–445.
- Kang, S. B. and Ikeuchi, K. (1997). Toward Automatic Robot Instruction from Perception - Mapping Human Grasps to Manipulator Grasps. *IEEE Transactions on Robotics and Automation*, 11:432–443.

- Kawasaki, H., Komatsu, T., and Uchiyama, K. (2002). Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II. *Transactions on Mechatronics*, 7(3):296–303.
- Kerdvibulvech, C. and Saito, H. (2008). Markerless Guitarist Fingertip Detection Using a Bayesian Classifier and a Template Matching For Supporting Guitarists. In *Proc. 10th International Conference on Virtual Reality*.
- Khansari-Zadeh, S. M. and Billard, A. (2011). Learning Stable Nonlinear Dynamical Systems With Gaussian Mixture Models. *Transactions on Robotics*, 27(5):943–957.
- Kimura, H., Fukuoka, Y., and Cohen, A. (2007). Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts. *International Journal of Robotics Research*, 26(5):475–490.
- Kjellstrom, H., Romero, J., and Kragic, D. (2008). Visual recognition of grasps for human-to-robot mapping. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3192–3199.
- Knoop, S., Vacek, S., and Dillmann, R. (2009). Fusion of 2D and 3D sensor data for articulated body tracking. *Robotics and Autonomous Systems*, 57(3):321–329.
- Kolb, D. (1984). *Experiential learning: experience as the source of learning and development*. Prentice Hall, Englewood Cliffs, NJ.
- Komkov, V. (1972). *Optimal Control Theory for the Damping of Vibrations of Simple Elastic Systems*. Lecture Notes in Mathematics, Vol. 253. Springer, Berlin, New York.
- Kroemer, O., Detry, R., Piater, J., and Peters, J. (2010). Combining Active Learning and Reactive Control for Robot Grasping. *Robotics and Autonomous Systems*, 58:1105–1116.
- Kroemer, O., Ugur, E., Oztop, E., and Peters, J. (2012). A Kernel-based approach to Direct Action Perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2605–2610, St. Paul, USA.
- Kulić, D., Takano, W., and Nakamura, Y. (2008). Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *The International Journal of Robotics Research*, 27(7):761–784.
- Lawson, C. L. and Hanson, R. J. (1974). *Solving Least Squares Problems*. Prentice Hall, Englewood Cliffs, NJ, USA.
- Levenberg, K. (1944). A Method for the Solution of Certain Non-Linear Problems in Least Squares. *Quarterly of Applied Mathematics*, 2:164–168.
- Li, Y. and Pollard, N. S. (2005). A shape matching algorithm for synthesizing humanlike enveloping grasps. In *humanoids*, pages 442–449.
- Ljung, L. (1999). *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, NJ, USA.
- Lloyd, B. A., Székely, G., and Harders, M. (2007). Identification of spring parameters for deformable object simulation. *Visualization and Computer Graphics, IEEE Transactions on*, 13(5):1081–1094.

- Lourakis, M. (2005). A brief description of the Levenberg-Marquardt algorithm implemented by levmar. *Foundation of Research and Technology*, 4:1–6.
- Majjad, R. (1997). Estimation of suspension parameters. In *IEEE International Conference on Control Applications*, pages 522–527.
- Markovsky, I. and Huffel, S. V. (2007). Overview of total least-squares methods. *Signal processing*, 87(10):2283–2302.
- Matsui, D., Minato, T., MacDorman, K. F., and Ishiguro, H. (2005). Generating Natural Motion in an Android by Mapping Human Motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3301–3308, Edmonton, Alberta, Canada.
- Miller, A. T., Knoop, S., Christensen, H. I., and Allen, P. K. (2003). Automatic Grasp Planning using Shape Primitives. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1824–1829, Taipei, Taiwan.
- Montesano, L., Lopes, M., Bernardino, A., and Santos-Victor, J. (2007). Learning Object Affordances: From Sensory Motor Coordination to Imitation. *Transactions on Robotics*, 24(1):15–26.
- Mouri, T., Kawasaki, H., Yoshikawa, K., Takai, J., and Ito, S. (2002). Anthropomorphic Robot Hand: Gifu Hand III. In *Proc. International Conference on Control, Automation and Systems*, pages 1288–1293.
- Napier, J. R. (1956). The prehensile movements of the human hand. *The Journal of bone and joint surgery*, 38-B:902–913.
- Nguyen, T. N. and Stephanou, H. E. (1990). A topological algorithm for continuous grasp planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 670–675.
- Ninness, B. and Gibson, S. (2001). The EM algorithm for Multivariable Dynamic System Estimation. In *IFAC International Workshop on Adaptation and Learning in Control and Signal Processing*.
- Oikonomidis, I., Kyriazis, N., and Argyros, A. A. (2010). Markerless and Efficient 26-DOF Hand Pose Recovery. In *Proceedings of 10th Asian Conf. Computer Vision*.
- Palm, R. and Iliev, B. (2006). Learning of grasp behaviors for an artificial hand by time clustering and Takagi-Sugeno modeling. In *IEEE International Conference on Fuzzy Systems*, pages 291–298.
- Pastor, P., Hoffmann, H., Asfour, T., and Schaal, S. (2009). Learning and Generalization of Motor Skills by Learning from Demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan.
- Paulignan, Y., Frak, V. G., Toni, I., and Jeannerod, M. (1997). Influence of object position and size on human prehension movements. *Experimental Brain Research*, 114(2):226–234.
- Pham, D. T. (2001). Stochastic methods for sequential data assimilation in strongly nonlinear systems. volume 129, pages 1194–1207.

- Pratt, J., Chew, C.-M., Torres, A., Dilworth, P., and Pratt, G. (2001). Virtual Model Control: An Intuitive Approach for Bipedal Locomotion. *International Journal of Robotics Research*, 20:129–143.
- Przybylski, M., Asfour, T., and Dillmann, R. (2011). Planning grasps for robotic hands using a novel object representation based on the medial axis transform. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1781–1788.
- Rand, M. K., Shimansky, Y. P., Hossain, A. B., and Stelmach, G. E. (2008). Quantitative model of transport-aperture coordination during reach-to-grasp movements. *Experimental Brains Research*, 188(2).
- Rao, K., Medioni, G., Liu, H., and Bekey, G. A. (1989). Shape description and grasping for robot hand-eye coordination. *Control Systems Magazine*, 9(2):22–29.
- Rehg, J. and Kanade, T. (1994). DigitEyes: Vision-Based Hand Tracking for Human-Computer Interaction. In *Proc. Workshop Motion of Non-Rigid and Articulated Bodies*, pages 16–22.
- Righetti, L. and Ijspeert, A. (2006). Programmable central pattern generators: an application to biped locomotion control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1585–1590.
- Romero, J., Feix, T., Kjellström, H., and Kragic, D. (2010). Spatio-temporal modeling of grasping actions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2103–2108, Taipei, Taiwan.
- Salisbury, J. K. (1983). *Kinematic and Force Analysis of Articulated Hands*. PhD thesis, Stanford University.
- Salisbury, J. K. and Roth, B. (1983). Kinematic and Force Analysis of Articulated Mechanical Hands. *Journal of Mechanisms, Transmissions and Automation*, 105(1):35–41.
- Santello, M., Flanders, M., and Soechting, J. F. (1998). Postural hand synergies for tool use. *Journal of Neuroscience*, 18:10105–10115.
- Schaal, S. (1997). Learning from Demonstration. In *Advances in Neural Information Processing Systems 9*, pages 1040–1046, Denver, USA.
- Schaal, S. and Atkeson, C. G. (1998). Constructive incremental learning from only local information. (8):2047–2084.
- Schaal, S., Peters, J., Nakanishi, J., and Ijspeert, A. (2005). Learning movement primitives. In *Robotics Research*, pages 561–572. Springer.
- Schaal, S., Sternad, D., Osu, R., and Kawato, M. (2004). Rhythmic arm movement is not discrete. *Nature neuroscience*, 7(10):1136–1143.
- Schlesinger, G. (1919). Ersatzglieder und Arbeitshilfen für Kriegsbeschädigte und Unfalverletzte. In *Der Mechanische Aufbau der Künstlichen Glieder*, pages 312–661. Springer, Berlin/Heidelberg.
- Schulz, S., Pylatiuk, C., and Bretthauer, G. (2001). A new ultralight anthropomorphic hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2437–2441.

- Sekimoto, M. and Arimoto, S. (2006). Experimental Study on Reaching Movements of Robot Arms with Redundant DOFs Based upon Virtual Spring-Damper Hypothesis . In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 562–567, Beijing, China.
- Serban, R. and Freeman, J. S. (2001). Identification and Identifiability of Unknown Parameters in Multibody Dynamic Systems. *Multibody System Dynamics*, 5(4):335–350.
- Shadmehr, R. and Wise, S. P. (2005). *Computational Neurobiology of Reaching and Pointing: A Foundation for Motor Learning*. MIT Press, Cambridge, MA, USA.
- Shadow Robot Company (2013). Shadow Dexterous Hand. Available online at <http://http://www.shadowrobot.com/products/dexterous-hand/>.
- Shukla, A. and Billard, A. (2012). Coupled dynamical system based arm-hand grasping model for learning fast adaptation strategies. *Robotics and Autonomous Systems*, 60:424–440.
- Smeets, J. B. J. and Brenner, E. (1999). A new view on grasping. *Motor Control*, 3:237–271.
- Smola, A. (2011). Support Vector Regression. *International Journal of Robotics Research*, 30:1229–1249.
- Steffen, J. F., Haschke, R., and Ritter, H. (2008). Towards Dextrous Manipulation Using Manifolds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Stein, T., Fischer, A., Boesnach, I., Köhler, H., Gehrig, D., and Schwameder, H. (2007). *Kinematische Analyse menschlicher Alltagsbewegungen für die Mensch-Maschine-Interaktion*. V. Aachen: Shaker.
- Stenger, B., Mendonca, P. R. S., and Cipolla, R. (2001). Model-based 3D tracking of an articulated hand. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 310–315.
- Stulp, F., Theodorou, E., Buchli, J., and Schaal, S. (2011). Learning to grasp under uncertainty. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5703–5708.
- Sujan, V. A. and Dubowsky, S. (2003). An Optimal Information Method for Mobile Manipulator Dynamic Parameter Identification. *IEEE/ASME Transactions on Mechatronics*, 2(2):215–225.
- Swevers, J., Ganseman, C., Tüekel, D. B., Schutter, J. D., and Brussel, H. V. (1997). Optimal Robot Excitation and Identification. *IEEE Transactions on Robotics and Automation*, 14(5):730–740.
- Taha, Z., Brown, R., and Wright, D. (1997). Modelling and simulation of the hand grasping using neural networks. *Medical engineering & physics*, 19(7):536–538.
- Tegin, J., Ekvall, S., Kragic, D., Wikander, J., and B.Iliev (2009). Demonstration-based learning and control for automatic grasping. In *Intelligent Service Robotics*, volume 2, pages 23–30. Springer, Berlin/Heidelberg.

- Triesch, J., Wieghardt, J., Maël, E., and v. d. Malsburg, C. (1999). Towards Imitation Learning of Grasping Movements by an Autonomous Robot. In *Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, GW '99, pages 73–84, London, UK. Springer.
- Tso, S. K. and Liu, K. P. (1996). Hidden Markov model for intelligent extraction of robot trajectory command from demonstrated trajectories. In *IEEE International Conference on Industrial Technology (ICIT)*, pages 294–298.
- Ude, A., Atkeson, C., and Riley, M. (2004). Programming Full-Body Movements for Humanoid Robots by Observation. *Robotics and Autonomous Systems*, 47(2-3):93–108.
- Ude, A., Gams, A., Asfour, T., and Morimoto, J. (2010). Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Transactions on Robotics*, 26(5):800–815.
- Ugur, E., Sahin, E., and Oztop, E. (2012). Self-discovery of motor primitives and learning grasp affordances. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3260–3267.
- Uno, Y., Fukumura, N., Suzuki, R., and Kawato, M. (1995). A computational model for recognizing objects and planning hand shapes in grasping movements. *Neural Networks*, 8(6):839–851.
- Vahrenkamp, N. (2010). Simox - A lightweight simulation and motion planning toolbox for C++. Available online at <http://http://simox.sourceforge.net>.
- Vahrenkamp, N., Wieland, S., Azad, P., Gonzalez-Aguirre, D., Asfour, T., and Dillmann, R. (2008). Visual Servoing for Humanoid Grasping and Manipulation Tasks. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 406–412, Daejeon, Korea.
- Verheij, R., Brenner, E., and Smeets, J. B. J. (2012). Grasping Kinematics from the Perspective of the Individual Digits: A Modelling Study. *PLoS One*, 7.
- Vicon Motion Systems (2013). Vicon T-Series. Available online at <http://www.vicon.com/System/TSeries>.
- Vinayavekhin, P., Kudoh, S., and Ikeuchi, K. (2011). Towards an Automatic Robot Regrasping Movement based on Human Demonstration using Tangle Topology. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3332–3339, Shanghai, China.
- Wachter, S. and Nagel, H.-H. (1999). Tracking of Persons in Monocular Image Sequences. *Computer Vision and Understanding*, 74(3):174–192.
- Wada, Y. and Kawato, M. (1995). A theory for cursive handwriting based on the minimization principle. *Biological Cybernetics*, 73(1):3–13.
- Wampler, C. W. (1986). Manipulator Inverse Kinematic Solutions based on Vector Formulations and Damped Least Squares Methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 16:93–101.
- Winter, D. A. (2009). *Biomechanics and motor control of human movement*. John Wiley & Sons.



- 
- Wolpert, D. M. and Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329.
- Wu, J., Wang, J., and You, Z. (2010). An overview of dynamic parameter identification of robots. *Robotics and Computer-Integrated Manufacturing*, 26:414–419.
- Xperience Project (2011). Robots Bootstrapped through Learning from Experience. Available online at <http://www.xperience.org>.
- Yamashita, N. and Fukushima, M. (2001). On the rate of convergence of the Levenberg-Marquardt method. In *Topics in numerical analysis*, pages 239–249. Springer.
- Yaskawa Motoman Robotics (2013). Motoman-SIA20D Datasheet. Available online at <http://www.motoman.com/datasheets/SIA20D.pdf>.
- Young, P. (1981). Parameter Estimation for Continuous-Time Models - A Survey. *Automatica*, 17(1):23–39.
- Zhao, X., Huang, Q., Peng, Z., and Li, K. (2004). Humanoid Kinematics Mapping and Similiarity Evaluation based on Human Motion Capture. In *IEEE International Conference on Information Acquisition*, pages 426–431, Hefei, China.
- Zivkovic, Z. and Kroese, B. (2004). An EM-like algorithm for color-histogram-based object tracking. In *Proc. Int Conf. Computer Vision and Pattern Recognition*, pages 798–803, Washington D.C., USA.
- Zoellner, R., Asfour, T., and Dillmann, R. (2004). Programming by Demonstration: Dual-Arm Manipulation Tasks for Humanoid Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 479–484, Sendai, Japan.