

Cognition Rehearsed

Recognition and Reproduction of Demonstrated Behavior

Erik A. Billing



PHD THESIS, JANUARY 2012
DEPARTMENT OF COMPUTING SCIENCE
UMEÅ UNIVERSITY
SWEDEN

Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden

billing@cs.umu.se
www.cs.umu.se/personal/erik-billing

Copyright © 2011 by authors

Except Paper I, © 2010 INSTICC Press

Paper II, © 2008 IEEE

Paper III, © 2010 Springer Verlag

Paper IV, © 2011 Springer Verlag

Paper V, © 2010 IEEE

ISBN 978-91-7459-349-5

ISSN 0348-0542

UMINF 11.16

December 21, 2011

Front cover by Johan Billing, Mena Abd Mohammed, and Pär Andersson.
Printed by Print & Media, Umeå University, 2011.

Abstract

The work presented in this dissertation investigates techniques for robot *Learning from Demonstration (LFD)*. LFD is a well established approach where the robot is to learn from a set of demonstrations. The dissertation focuses on LFD where a human teacher demonstrates a behavior by controlling the robot via teleoperation. After demonstration, the robot should be able to reproduce the demonstrated behavior under varying conditions. In particular, the dissertation investigates techniques where previous behavioral knowledge is used as bias for generalization of demonstrations.

The primary contribution of this work is the development and evaluation of a semi-reactive approach to LFD called *Predictive Sequence Learning (PSL)*. PSL has many interesting properties applied as a learning algorithm for robots. Few assumptions are introduced and little task-specific configuration is needed. PSL can be seen as a variable-order Markov model that progressively builds up the ability to predict or simulate future sensory-motor events, given a history of past events. The knowledge base generated during learning can be used to control the robot, such that the demonstrated behavior is reproduced. The same knowledge base can also be used to recognize an on-going behavior by comparing predicted sensor states with actual observations. Behavior recognition is an important part of LFD, both as a way to communicate with the human user and as a technique that allows the robot to use previous knowledge as parts of new, more complex, controllers.

In addition to the work on PSL, this dissertation provides a broad discussion on representation, recognition, and learning of robot behavior. LFD-related concepts such as *demonstration*, *repetition*, *goal*, and *behavior* are defined and analyzed, with focus on how bias is introduced by the use of behavior primitives. This analysis results in a formalism where LFD is described as transitions between information spaces. Assuming that the behavior recognition problem is partly solved, ways to deal with remaining ambiguities in the interpretation of a demonstration are proposed.

The evaluation of PSL shows that the algorithm can efficiently learn and reproduce simple behaviors. The algorithm is able to generalize to previously unseen situations while maintaining the reactive properties of the system. As the complexity of the demonstrated behavior increases, knowledge of one part of the behavior sometimes interferes with knowledge of another parts. As a result, different situations with similar sensory-motor interactions are sometimes confused and the robot fails to reproduce the behavior.

One way to handle these issues is to introduce a context layer that can support PSL by providing bias for predictions. Parts of the knowledge base that appear to fit

the present context are highlighted, while other parts are inhibited. Which context should be active is continually re-evaluated using behavior recognition. This technique takes inspiration from several neurocomputational models that describe parts of the human brain as a hierarchical prediction system. With behavior recognition active, continually selecting the most suitable context for the present situation, the problem of knowledge interference is significantly reduced and the robot can successfully reproduce also more complex behaviors.

Sammanfattning

Den här avhandlingen presenterar en undersökning av metoder för robotinlärning från demonstrationer (LFD). LFD är en väl etablerad teknik för att lära robotar nya beteenden. Avhandlingen fokuserar på LFD där en mänsklig lärare fjärrstyr roboten medan motorkommandon och sensoravläsningar spelas in. Efter demonstrationen ska roboten kunna reproducera beteendet under varierande förhållanden. Möjligheten att använda tidigare motorisk kunskap för att tolka demonstrationen undersöks. Denna information kan underlätta generalisering av demonstrationen, så att beteendet kan reproduceras även när förhållandena i omgivningen förändrats.

Det huvudsakliga vetenskapliga bidraget i den här avhandlingen är en semireaktiv algoritm för LFD benämnd *Predictive Sequence Learning (PSL)*, samt en serie utvärderingar av denna. PSL har flera intressanta egenskaper när den appliceras som metod för LFD. PSL kräver endast begränsad anpassning till nya applikationer och få antaganden introduceras. Algoritmen kan ses som en Markovmodell som anpassar tillståndsrymden efter det data som den tränas på. Genom träning genereras en modell som kan användas för att predicera eller simulera sensor- och motortillstånd som spelats in vid demonstrationer. Modellen kan användas för att kontrollera roboten så att det demonstrerade beteendet reproduceras. Modellen kan också användas för att känna igen ett pågående beteende. Detta görs genom att predicerade sensortillstånd jämförs med observerade. Denna förmåga att känna igen beteenden är viktig för LFD, både som ett sätt att kommunicera med användaren men också som en teknik som möjliggör användandet av tidigare kunskap för att tolka demonstrationer.

Utöver arbetet med PSL presenteras en diskussion om representation, igenkänning och inlärning av robotars beteende. LFD-relaterade koncept som demonstration, repetition, mål och beteende definieras och analyseras, med fokus på hur förkunskap kan introduceras genom beteendepremitiv. Analysen resulterar i en formalism där LFD beskrivs i termer av övergångar mellan informationsrymder. Flera sätt att hantera tvetydigheter i tolkningen av demonstrationer föreslås.

Utvärderingen av PSL visar att algoritmen är användbar som en reglermetod för robotar. PSL kan på ett effektivt sätt representera och reproducera enklare beteenden, samt generalisera till nya situationer. För mer komplexa beteenden ökar dock risken att delar av den genererade modellen stör andra delar, och det inlärdade beteendet kan inte reproduceras på ett korrekt sätt. Ett sätt att hantera detta problem är att introducera ett kontextlager. Kontextlagret kan stödja PSL genom att aktivera de delar av modellen som hör till den aktuella kontexten, medan övriga delar inhiberas. Den prediktiva modellen kan användas för att beräkna hur den aktuella situationen är förenlig med

olika kontexter. Roboten kan på så vis automatiskt aktivera den kontext som bäst passar den aktuella situationen. Denna metod är inspirerad av flera beräkningsmässiga modeller av nervsystemet vilka beskriver hjärnan som ett hierarkiskt prediktionssystem. När kontextlagret används minskar risken att delar av modellen stör andra delar, och roboten kan framgångsrikt reproducera mer komplexa beteenden.

Preface

This thesis consists of an introduction, an overview of relevant research, and the following seven articles.

- Paper I Erik A. Billing. Cognitive Perspectives on Robot Behavior. In *Proceedings of the Second International Conference on Agents and Artificial Intelligence, Special Session on Languages with Multi-Agent Systems and Bio-Inspired Devices*, p. 373–382. INSTICC Press. Valencia, Spain, January 22–24, 2010.
- Paper II Erik A. Billing and Thomas Hellström. Behavior Recognition for Segmentation of Demonstrated Tasks. In Vladimír Mařík, Jeffery M. Bradshaw, Joachim Meyer, William A. Gruver, and Petr Benda (Eds.), *Proceedings of IEEE SMC International Conference on Distributed Human-Machine Systems*, p 228–234. IEEE. Athens, Greece. March 9–12, 2008.
- Paper III Erik A. Billing and Thomas Hellström. A Formalism for Learning from Demonstration. *Paladyn: Journal of Behavioral Robotics*. 1:1, p. 1–13. Versita, co-published with Springer Verlag. March 2010.
- Paper IV Erik A. Billing, Thomas Hellström, and Lars-Erik Janlert. Predictive learning from demonstration. In Joaquim Filipe, Ana Fred, and Bernadette Sharp (Eds.), *Agents and artificial Intelligence: Revised Selected Papers*, p. 186–200. Springer Verlag. Communications in Computer and Information Science, 129. 2011.
- Paper V Erik A. Billing, Thomas Hellström, and Lars-Erik Janlert. Behavior Recognition for Learning from Demonstration. In *Proceedings of IEEE International Conference on Robotics and Automation*, p. 866–872. IEEE. Anchorage, Alaska, May 3–8, 2010.
- Paper VI Erik A. Billing, Thomas Hellström, and Lars-Erik Janlert. Robot Learning from Demonstration using Predictive Sequence Learning. To appear in A. Dutta (Ed.), *Robotic Systems - Applications, Control and Programming*. InTech. 2011.
- Paper VII Erik A. Billing, Thomas Hellström, and Lars-Erik Janlert. *Simultaneous Control and Recognition of Demonstrated Behavior*. Technical Report, UMINF 11.15. Department of Computing Science. Umeå University. Sweden. 2011.

Additional work

Minor additional contributions can be found in the following papers by the author.

1. Erik A. Billing. *Simulation of Corticospinal Interaction for Motor Control*. Master Thesis. Cognitive Science Programme, Department of Integrative Medical Biology, Umeå University, Umeå, Sweden. 2004.
2. Erik A. Billing and Thomas Hellström. Behavior and Task Learning from Demonstration. In *Proceedings of the 23rd Annual workshop of the Swedish Artificial Intelligence Society (SAIS06)*, p. 151. Umeå, Sweden. May 10-12, 2006.
3. Erik A. Billing. *Representing Behavior - Distributed theories in a context of robotics*. Technical Report, UMINF 07.25. Department of Computing Science. Umeå University. Sweden. 2007.
4. Erik A. Billing. *Cognition Reversed - Robot Learning from Demonstration*. Licentiate Thesis. Department of Computing Science. Umeå University. Sweden. 2009.
5. Erik A. Billing, Thomas Hellström, and Lars-Erik Janlert. Model-free Learning from Demonstration. In *Proceedings of the Second International Conference on Agents and Artificial Intelligence*, p. 62-71. INSTICC Press. Valencia, Spain, January 22–24, 2010.
6. Erik A. Billing. *Achilles' heel of cognitive science*. Technical Report, UMINF 11.14. Department of Computing Science. Umeå University. Sweden. 2011.

Path to dissertation

When I started my PhD studies in 2006 I was convinced that robots able to act and learn like humans do were science fiction and not a realistic research topic. I had taken what I saw as a mature perspective on artificial intelligence, aligning to a weak AI perspective. During my undergraduate studies at the Cognitive Science Program¹, I was taught that cognition is about how humans, animals and artificial systems perceive information, process it and finally respond with some output or action. Since I had not even seen computers able to solve the perception problem in any way comparable to humans' and animals' perceptual abilities, I could not see how we could even approach the problems of implementing human-like information processing and action abilities in robots. Of course there were many specific applications where robots were successful, but my interest lay, and still lies, in a general understanding of cognition. In this context, robot learning appeared as one area where general solutions were still in focus.

I directed my attention to robot *Learning From Demonstration (LFD)*, where the robot is to learn from a set of examples or demonstrations. I focused on scenarios where a human teacher is controlling the robot pupil via teleoperation. In this context, a *demonstration* is a sequence of sensor readings and motor commands issued by the teacher during execution of the desired behavior. While this kind of scenario may not resemble the way humans teach each other, they constitute practically useful settings generalizable to many kinds of robots.

I was initially interested in how behavior should be represented in robots. When reviewing the literature on intelligent robotics and robot learning, leading up to Paper I, I had problems to find a clear consensus on what methods to use. Many of the proposed methods appeared to fit the particular application well, but it was difficult to get an understanding of which methods that would work best in the general case. Together with my supervisor Thomas Hellström², I decided to direct my attention to approaches that used so called behavior primitives or skills as a method for LFD. A *behavior primitive* is a simple controller that can be combined with other controllers to form more complicated behaviors. Without specifying how each primitive was to be implemented, we could still reason about how they could be combined. If we could create a system able to combine primitives on several levels, such that combined skills could constitute primitives for even more complex behaviors, a hierarchical structure would emerge able to gradually increase the robot's knowledge.

¹ Cognitive Science Program, Department of Psychology, Umeå University, Umeå, Sweden

² Assoc. Prof. Thomas Hellström, Department of Computing Science, Umeå University, Umeå, Sweden

We realized the importance of behavior recognition, i.e., that the robot must be able to recognize some part of a demonstration corresponding to a known behavior primitive. We developed and evaluated three techniques for behavior recognition, presented in Paper II. During this work we realized that behavior recognition was a very hard problem. Even simple demonstrations could be manifestations of a great variety of different behaviors. Small changes in the environment or the controller could result in a completely different sequence of sensory-motor events constituting the demonstration. Me and Thomas Hellström put a lot of work into analyzing and formalizing these issues, resulting in Paper III.

The conclusion was that some assumptions (biases) had to be introduced to make learning possible. Even though this was an obvious conclusion for anyone with some experience in machine learning, I couldn't help but finding it really annoying. If we have to introduce information about the behavior prior to learning, then what good does learning do? One could of course argue that we must rely on some very basic assumptions, applicable in many situations and behaviors, but this wasn't how it was done in practice. The kind of assumptions that we, and many other researchers in the field, introduced were specific things, like what aspects of objects were relevant, how positions of the robot and objects in the environment should be represented, and with which granularity the sensors could perceive the world. All these assumptions are typical examples of ontological information that is necessary for any knowledge representation. It seemed to me that what we did was building more and more information into the robot until the interpretation became obvious. This was in direct conflict with the kind of incremental learning that we aimed for when using behavior primitives.

In the middle of all this, a colleague, Daniel Sjölie³, directed me to a book called *On Intelligence* by Jeff Hawkins. For me, this book became the first step into a field of research investigating high level computational aspects of the brain. I had been working with computational neuroscience for my Master Thesis⁴, and was happy to find a book that actually put knowledge from both neuroscience and computing science together. About the same time, Ben Edin⁵, supervisor for my Master Thesis, directed me to the work by Brandon Rohrer at Sandia National Laboratories. Both the work by Rohrer and Hawkins focus less on *where* in the brain it happens, and more on *how* it happens. Two things in Hawkins's book really caught my attention.

1. Cortex is primarily a memory system
2. The whole cortex performs one and the same basic computation, referred to as the *common cortical algorithm (CCA)*

If the idea about CCA is right, it should be possible to formulate it in computational terms and implement it in a computer, allowing robots to learn like humans and other animals do. While the brain does not work like a computer and a computer may not

³ Daniel Sjölie, Department of Computing Science, Umeå University, Umeå, Sweden

⁴ Erik A. Billing. *Simulation of Corticospinal Interaction for Motor Control*. Master thesis. Department of Integrative Medical Biology, Umeå University, Sweden

⁵ Prof. Ben Edin, Department of Integrative Medical Biology, Umeå University, Umeå, Sweden

be an efficient platform for implementing the kind of computations performed by the brain, the brain does learn without a programmer telling it what is important and I got convinced that the best way to figure out how to do the same in robots is to understand how the brain works.

During autumn 2008 and spring 2009 I studied several models of the brain which resulted in an overview constituting large parts of the introduction chapters to my Licentiate thesis⁶. Inspired by Rohrer's work on modeling motor control, we also developed the algorithm *Predictive Sequence Learning (PSL)* which forms the basis for papers IV to VII of this dissertation. PSL is a dynamical temporal difference algorithm that introduces very few assumptions into learning. In the work presented in Paper IV, PSL was applied to an LFD-problem, learning to control a Khepera miniature robot. Based on PSL, we also developed two algorithms for behavior recognition. The new algorithms were compared with our previous work on behavior recognition. The results are presented in Paper V. The work with Paper IV and Paper V showed that PSL could be used both as a controller and as a method for behavior recognition, but also revealed a number of problems and limitations with the algorithm.

In December 2009, I presented my Licentiate thesis and during the spring that followed we explored several ways to continue the work on PSL. In order to allow larger knowledge bases, I spent some time on implementing a version of PSL that could store the knowledge base in a standard relational database. This implementation did however prove to be too slow to be useful for robotic applications. Almost half a year was spent on applying PSL to a reinforcement learning task. The idea was to use the growing knowledge base of PSL as basis for generalizing rewards, potentially creating a system that dynamically constructed a state space suitable for the particular task. This proved to be much more difficult than expected and also directed me away from LFD, that was the main focus of my dissertation. We therefore decided to cancel this direction and unfortunately I've not found the time to pick it up within the time of my PhD studies.

We also put work into a new version of PSL based on Fuzzy Logic (presented in papers IV and VII). The new version handles data with many dimensions in a better way than the original algorithm, which made it possible to scale up the evaluation environment from the Khepera robot to a human size Kompai robot. While all results presented in papers VI and VII are taken from the simulated environment, experiments on the physical robot were made parallel to this work. We were however not able to finish the experiments on the physical robot in time for this dissertation.

Inspired by the neurocomputational models reviewed in my Licentiate thesis, we also explored the possibility to create a hierarchical system based on the original PSL algorithm. While a complete implementation of such an architecture has not been done within the timespan of this dissertation, several components have been implemented and evaluated. In Paper VII, a context layer for PSL is introduced. The behavior recognition abilities of PSL is used to continuously select the most suitable context while the robot is driving. The context layer provides bias to PSL by activating some parts of the knowledge base, while inhibiting other parts. The architecture presented

⁶ Erik A. Billing. *Cognition Reversed - Robot Learning from Demonstration*. Licentiate Thesis. Department of Computing Science. Umeå University. Umeå. Sweden. 2009.

in Paper VII could potentially be extended with a second instance of PSL running at the context level, further supporting selection of suitable contexts. Such a two-layer architecture could be further extended with more layers, producing a dynamic hierarchical system making predictions at multiple levels of abstraction (see Chapter 3 for details).

The results presented in Paper VII are promising and I am now finishing this dissertation with a feeling that I want to do so much more. I want to fully explore the possibilities of the kind of learning architecture that this dissertation embraces. In a couple of years, if I look back on this thesis, the text will probably appear different to me. My brain may have rehearsed the arguments yet a number of times, hierarchical learning systems may not appear as thrilling as they do now, and I will hopefully see their limitations much clearer. I may use different knowledge to interpret these words, and they may mean different things to me, than they do now. If that is so, I will be happy.

Acknowledgements

Wednesday, June first 2005, I made a mistake. I wrote and failed the exam on the course Intelligent Robotics at the Department of Computing Science, Umeå University. I had not studied enough, obviously. Even though I found the subject very interesting, I could not see how it would ever contribute to my future career. The failure was a close cut, and the course responsible Thomas Hellström gave me the opportunity to do a project work rather than taking the re-exam. The project went well, and when I was finished, Hellström asked me if I would like to become PhD student. And I did.

With all my heart, I now, more than six years later, express my great gratitude to my supervisor Assoc. Prof. Thomas Hellström for believing in me and giving me this opportunity to become a PhD. Thank you for all the long discussions, our many arguments, and for being there when I needed you.

Even though Hellström was the one who pulled me into PhD studies and was most present in my work for the first years, my secondary supervisor Prof. Lars Erik Janlert has also been an invaluable mentor during my PhD studies. Thank you for providing guidance and detailed comments on my work. Thank you for many interesting discussions, especially during the last part of my PhD studies. But most of all, thank you for always trusting in me and providing a sense of calm when needed. And thank you for pulling me into Swecog.

The National Graduate School of Cognitive Science (Swecog) has been a very important platform for me. I would like to thank Christian Balkenius, Nils Dahlbäck, and the other members of Swecog, for providing a very inspiring community for discussion and reflection which have helped me enforce the cognitive direction of my research. I also acknowledge Brandon Rohrer for valuable input to this work.

I thank Ola Ringdahl and Johan Tordsson who all since I came to the department have been close colleagues, helping me out with all these small, daily things that are so important. I would also like to thank Daniel Sjölie who provided some of the most valuable directions for the work presented in this thesis, and Benjamin Fonooni who made important work on the software platform used for papers VI and VII. Thanks also go to Stefan Holmgren, Lennart Edblom, and Lena Kalin Westin for giving me the opportunity to teach as much as I have during these years. It has been hard at times, but very rewarding. I also express my gratitude to all other colleagues at the department for providing a warm environment which makes it easy to go to work in the morning. Special thanks goes to Tommy Eriksson, Roland Johansson, Yvonne Löwstedt, Anne-Lie Persson, Inger Sandgren, and the department's support group, for

Acknowledgements

always keeping a positive attitude and helping out with all the practical things.

Finally, thanks goes to my dear Mena, my friends, my parents, and my brother. Thank you for dragging me out of the office, and for letting me stay at times. Thank you for helping me forget work when I need to, and for reminding me that there are other things than robots worth exploring. Without you, this dissertation would probably be more comprehensive, but I would not be wiser.

Contents

1	Introduction	1
2	Learning from demonstration	5
2.1	Level of imitation	6
2.2	Control	7
2.3	Recognition	9
3	Hierarchical models for learning	13
3.1	Motivation for hierarchies	14
3.2	Hierarchical predictive learning	16
4	Summary of articles	21
5	Contributions	23
	Paper I	33
	Paper II	47
	Paper III	59
	Paper IV	77
	Paper V	97
	Paper VI	109
	Paper VII	129

CHAPTER 1

Introduction

Robots are more present in our society than ever before. The first autonomous cars are now driving on public streets (Taylor III, 2011), the first humanoid robots are helping customers in shopping malls (Pal Robotics, 2011) and robots are becoming increasingly important for industry (Bischoff & Guhl, 2009). It is challenging to develop robots for several reasons. Robots are often aimed for applications that require high precision, handling of heavy loads, and are typically expected to execute tasks faster and more reliably than humans. Most tasks require that actions are executed in relation to the environment, in a safe way. Furthermore, the robot is expected to act on its own, without being directly controlled by a human user. To accomplish this, the robot is given sensors, actuators, and a computational unit. A computer program, referred to as a *controller*, reads and processes information from the sensors and controls the robot by sending signals to the actuators. The work presented in this dissertation is concerned with how to design controllers for robots acting in an everyday changing environment.

A controller π is defined as a mapping from a state $x \in X$ to an action $u \in U$:

$$\pi : X \rightarrow U \tag{1.1}$$

The state space X comprises all information necessary to select an action from the action space U . What information that is necessary depends on the particular behavior, but also on the robot's physical properties and its set of sensors and actuators. The current state x_t is defined by the use of information from sensors, mapping physical measures to a sensor state $y_t \in Y$, at time t .

Robots for special purposes, like lawn mowers and vacuum cleaners, are becoming common products. More flexible, multi-purpose, robots are however still far from the market. Robots could potentially be of great support in our daily lives; at work, and in our homes. Supporting us when we grow old, or serving as fun and interesting toys to play with as kids. Research on robots is also one way to better understand ourselves. Robots can be used as models of humans and other animals, supporting research on how we perceive information and control our actions (Berthouze & Metta, 2005).

One of the things that still prevent robots from entering everyday environments, like homes and office areas, are robots' limited ability to adapt to the environment. In an engineered environment, like a factory floor, the robot can be perfectly tuned to the application at design time, or through an iterative process where the robot is tested

and modified by the developers. However, in most other environments, the developers do not have complete information about the environment in which the robot is going to be used. For example, a robot that is to support a person by fetching things in that person's home, must know where he or she usually places things. This kind of information can only to limited extent be introduced at design time and the robot must therefore be able to store and use information gained from interaction with the environment in order to change π . We say that the robot must be able to *learn*.

Robot learning is not only about adapting an existing controller to a particular environment, but also about creating new behavior. The user may for example want his robot to also place things back at the right place in the apartment. In this case, the human user must be able to describe the desired behavior for the robot. There are at least two major approaches to robot learning. The robot can try out new things by itself, while the human gives feedback in terms of reward and punishment. This kind of learning is called *Reinforcement Learning (RL)*. Alternatively, the teacher may demonstrate the desired behavior to the robot. This kind of learning is called *Learning from Demonstration (LFD)* and is the primary focus of the work presented in this dissertation. The research problem of LFD can be formulated as how to represent information gained from demonstrations in such a way that the robot can reproduce the demonstrated behavior under varying conditions.

The term *behavior* is used to denote an agent's actions in relation to the environment and the term *demonstration* is used to refer to the information gained from the teacher showing how to execute a particular behavior. The teacher often has to demonstrate a behavior several times in order to allow the robot to generalize the behavior to new situations. The question of how a set of demonstrations should be generalized is central in LFD and is the main research question investigated in this dissertation.

In order to generalize demonstrations, bias is needed. That is, some basis on which the robot can choose one generalization over another. This very general claim, illustrated by the "no free lunch" theorems (Wolpert & Macready, 1997), applies not only to LFD, but to any learning or optimization system. In the context of robot learning, the "no free lunch" can be interpreted as an argument that it is not possible to create a general learning system that is always better than another. This can be seen as a good argument for conducting research on LFD only in limited domains, where domain specific knowledge can be introduced into the system. The work presented here focuses however on general approaches to LFD. The main thesis of this work is that such a general approach to learning is both possible and desirable, with the goal of showing why, and how, general learning can be achieved.

The techniques for LFD proposed in this dissertation use previous knowledge, gained from earlier learning sessions, as bias in future learning. Such an approach changes the generalization biases as learning progresses and allows the robot to progressively learn more complex behaviors. This ability to learn by the use of existing skills, possessed by humans and many animals, is illustrated by the *zone of proximal development*. This notion was introduced by Vygotsky (1978) in an argument against the use of standardized tests as a measure of students' intelligence. Vygotsky argues that a better gauge of intelligence is obtained by contrasting the results of students solving problems with, and without, guidance from others. The basic idea is that

learning can only take place when the task is not too easy and not too hard, but within the zone of proximal development. Taking a pupil through the zone of proximal development is called *scaffolding*, a term that has also become popular in robot learning (Berk & Winsler, 1995; Otero et al., 2008). In a scaffolded learning process, the pupil takes active part by exploiting new solutions based on known skills, while the teacher is supporting, scaffolding, the learning environment such that the pupil is able to complete the task. As learning progresses, teacher support is gradually reduced until the pupil can complete the task by itself.

The idea of using the result from previous learning sessions as basis for future learning is also present within the machine learning community, for example in form of *learning to learn* (e.g. Thrun & Pratt, 1998). These techniques aim to represent knowledge gained from learning such that it increases the performance of future learning. In the field of LFD, one common approach that implements this idea is the use of so called *behavior primitives* or *skills* (Fod et al., 2002; Matarić, 2002; Nakaoka et al., 2003; Nicolescu, 2003; Peters II et al., 2003; Bentivegna, 2004; Koenig & Matarić, 2006). A behavior primitive is a pre-programmed or previously learned controller that can execute a behavior, or some part of a behavior. A demonstration is matched with known primitives, transforming LFD into the problem of selecting a set of primitive controllers that can produce the demonstrated behavior. This process can be divided into three activities:

1. **Behavior segmentation** where a demonstration is divided into smaller segments.
2. **Behavior recognition** where each segment is associated with a primitive controller.
3. **Behavior coordination**, referring to identification of rules or switching conditions for how the primitives are to be combined.

These three activities were identified during the work on Paper III and remain central through most of the work presented in this dissertation. Specifically, the problem of *behavior recognition* is studied in detail. Behavior recognition can be seen as a classification problem of sequential data and is, just like the original generalization problem, in need of bias. While many solutions exist for specific controllers, a system able to recognize learned behaviors needs a generic solution to the problem of behavior recognition. One approach that may provide a general solution is to use a forward model (predictor) in combination with the inverse model (controller). The last four papers included in this dissertation comprise an investigation of one method along these lines. We call the proposed algorithm *Predictive Sequence Learning (PSL)*.

An introduction to LFD is given in Chapter 2. Chapter 3 introduces hierarchical models for LFD. A summary of the seven papers included in this dissertation is found in Chapter 4 and primary contributions of presented work are summarized in Chapter 5.

CHAPTER 2

Learning from demonstration

Successful *Learning from Demonstration (LFD)* requires that, given a set of demonstrations, a controller is generated such that the robot can reproduce the demonstrated behavior under varying conditions. This generalization process is difficult to formalize since it is often far from obvious how a particular set of demonstrations should be generalized. The relevance of different features in the demonstrations depend on how the behavior is demonstrated, what the purpose of the behavior is, and what sensors and actuators the robot has. It is therefore difficult to provide a precise formulation of how to generalize a demonstration, or a set of demonstrations. The notion of behavior is often used in a very general sense to describe some action in response to stimuli (e.g. Arkin, 1998) where it is basically up to the human teacher to freely decide whether a certain robot behavior is successful or not.

One way to structure the research field of LFD was made by the formulation of four questions of imitation learning: *what-to-imitate*, *how-to-imitate*, *who-to-imitate*, and *when-to-imitate* (Alissandrakis et al., 2002). The first question, *what-to-imitate*, was originally introduced in a classical work by Nehaniv & Dautenhahn (1999):

An action or sequence of actions is a successful component of imitation of a particular action if it achieves the same subgoal as that action. An entire sequence of actions is successful if it successively achieves each of a sequence of abstracted subgoals.

In other words, successful LFD requires that the goal of the demonstrated behavior be identified. In robotics, the formulation of a goal is often not trivial since it relies on background knowledge. Even if the goal state itself may be easily identified, for example a particular location in the environment, we usually demand that the robot be able to reach that location in a particular way. We may implicitly introduce the requirement that the robot be able to reach the target location within a certain time, and without hitting walls or objects on the way. It appears that there is no obvious limit to the amount of background knowledge required and it is therefore difficult to directly use a human's description of a goal as basis for the robot's behavior.

The second question, *how-to-imitate*, captures the problem of reproducing the behavior. Even if a suitable level of imitation is selected and the goal is identified, it is often not trivial to generate a controller that fulfills the goal. This problem is critical when the pupil has a different body structure. In this case, the teacher's actions

have to be transformed to corresponding actions for the robot pupil, introducing the *correspondence problem* (Nehaniv & Dautenhahn, 1999).

Who and when to imitate are the questions of selecting another agent that would be valuable to imitate and selecting the right time for imitation, respectively. Humans and animals do not just go around and imitate others all the time, but are able to identify parts of another agent's behavior that are valuable to copy. Parts of these problems are to identify a beginning and an end of demonstrations. Another related problem is to temporally align demonstrations such that common features can be identified even when the length of demonstrations varies. One common technique for temporal alignment is *dynamic time warping* (Myers & Rabiner, 1981), for example applied to kinesthetic demonstrations of a chess-piece moving task (Calinon et al., 2007).

Behaviors can be demonstrated to a robot in many different ways. Argall et al. (2009) outline four types of demonstrations: A direct recording of motor commands and sensor readings is referred to as an *identity record mapping*. In this case, the robot is controlled via tele-operation or by physically moving the robot's limbs (kinesthetic teaching). An external observation, e.g. a video recording of the teacher, is called a *non-identity record mapping*. This type of demonstrations poses a difficult sensing problem of detecting how the teacher has moved, but also allows much more flexible demonstration settings. The teacher may have a body identical to that of the pupil (*identity embodiment*) or a body with a different structure (*non-identity embodiment*). The latter case introduces the *correspondence problem* mentioned above. The work presented in this dissertation focuses on LFD via tele-operation. Sensor data and motor commands are recorded while a human teacher demonstrates the desired behavior by tele-operating the robot, producing demonstrations with identity in both record mapping and embodiment. In papers II to V, a miniature Khepera robot (K-Team, 2007) is used. The robot is controlled using a keyboard while motor commands and sensor readings are recorded. In papers VI and VII, a human size Kompai robot (Robosoft, 2011) in a simulated apartment environment is used. This robot is controlled using a joystick while motor commands and sensor readings are recorded in a similar way as with the previous work.

2.1 Level of imitation

One part of solving the what-to-imitate question is to identify suitable levels of abstraction at which the behavior is imitated. Stressing the hierarchical structure of behavior, Byrne & Russon (1998) identifies two distinct levels. The first level, corresponding to copying of the action sequence, is called *action-level imitation*. Actions may be executed in relation to stimuli, but with a fixed sequential structure. The second level, called *program-level imitation*, corresponds to imitation where the exact sequence of actions varies, while the overall structure of the behavior is copied. A set of demonstrations of a behavior that should be imitated at the action level is expected to have a fairly linear variability, with common statistical features. In contrast, demonstrations of a behavior at the program level may differ drastically considering sequence of actions, making it much harder to extract common features among sev-

eral demonstrations. In these situations it is often necessary to introduce high level knowledge about the behavior, often leading to specialized systems directed to LFD in limited domains.

A third level, the *effect-level imitation*, was introduced by Nehaniv & Dautenhahn (2001) in order to better describe imitation between agents with dissimilar body structures. With an effect level imitation, the imitation may look very different, both in terms of executed actions and their structure. An effect-level imitation is regarded successful if the produced effects on the environment, or the agent's relation to the environment, matches thaws of the demonstration.

Demiris & Hayes (1997) proposed three slightly different levels: 1) *basic imitation* with strong similarities to the notion of action-level imitation, 2) *functional imitation* that best corresponds to effect-level imitation and 3) *abstract imitation* that represents coordination based on the presumed internal state of the agent rather than the observed behavior. Demiris and Hayes give the example of making a sad face when someone is crying. In cases like this, it is clear that quite specific external information is required to draw the connection between the demonstration and the imitation, or between two sequences of actions that the human would argue are demonstrations of the same behavior.

The quality of an imitation at a specific imitation level, or a combination of imitation levels, has been formalized as a *metric of imitation*. The metric of imitation is defined as a weighted sum over all strategy-dependent metrics on all imitation levels (Billard et al., 2003). A *strategy* should be understood as an assumption of what is relevant in the demonstrated behavior. For example, a) to move a specific object, b) to move the objects in a specific direction, c) to move the objects in a specific sequence, d) to perform a specific gesture. The approach takes the perspective that the most frequent features of demonstrations are the most important and selects a strategy with optimal agreement among demonstrations. The metric of imitation was originally demonstrated on a manipulation task with a humanoid robot and has later been applied to a number of LFD applications. With focus on the correspondence problem, Alissandrakis et al. (2005) propose an approach to imitation of manipulation tasks. The what-to-imitate problem is approached by maximizing trajectory agreements of manipulated objects, using several different metrics. Some metrics encoded absolute trajectories while other metrics encoded relative object displacement and the relevant aspects of the behavior were in this way extracted as the common features in the demonstration set.

2.2 Control

Assuming that a suitable level of imitation has been identified, that the correspondence problem is solved, and that beginnings and ends of demonstrations have been found, we are left with the problem of deriving a controller, also referred to as control policy or forward model, π (Equation 1.1). The current state x_t is taken to be the determinant of action, which implies that the state must satisfy the Markov assumption. The Markov assumption states that given the state-action pair (x_t, u_t) , x_t is independent of

x_j for all $j < t < i$. In other words, (x_t, u_t) must encapsulate all information available to predict the future state x_i optimally. Argall et al. (2009) divide methods for control policy derivation into three classes:

1. **Mapping functions** use the demonstration to directly approximate a mapping from underlying states to actions.
2. **System models** use the demonstration to derive a model of the world dynamics. The model is often combined with a reward function that specifies the value of being in a certain model state, or taking an action in a certain state.
3. **Plans** use the demonstration to identify a set of pre- and post-conditions for each action. A sequence of actions can then be planned using a model of the state dynamics. The approach is often used together with additional user feedback.

As mentioned in the introduction, the information required by the controller varies with what kind of behavior it is to produce. It is therefore difficult to define a state space X suitable for any behavior. A large X , comprising very much information about the world, will introduce sensing problems when the robot is to identify the current state. Conversely, a small state space will limit the range of behaviors that the robot is able to learn. π is therefore often redefined as a function from the most recent observation $y_t \in Y$, or the agent's history of sensor and motor experiences $\eta_t = (e_1, e_2, \dots, e_t)$, where $e_i = (u_{t-1}, y_t)$:

$$u_t = \pi(y_t) \tag{2.1}$$

$$u_t = \pi(\eta_t) \tag{2.2}$$

Both Equation 2.1 and 2.2 are typical examples of mapping functions. They have the advantage that X is not explicitly represented and less prior assumptions are introduced into the system. Approaches based on system models or plans are however using different kinds of world representations. These approaches have the advantage that complex behavior can be represented more efficiently than possible with mapping functions, but usually require a state representation that is partly predefined. See Paper III and the work by Argall et al. (2009) for longer discussions.

The present work primarily investigates techniques associated with mapping functions. In this category, a number of classification and regression approaches can be found. Billard & Hayes (1999) use a recurrent neural network trained with Hebbian learning to encode control policies for mobile robots. Hovland et al. (1996) use a *Hidden Markov Model (HMM)* trained from human demonstrations to encode a controller for an assembly task. More recently, Calinon & Billard (2005) apply an HMM to encode gestures of a humanoid robot. Another technique that has recently become popular is to encode controllers with *Gaussian Mixture Models (GMM)*. Calinon et al. (2007) applies a mixture of Gaussian/Bernoulli distributions to a chess-piece moving task for a humanoid robot. GMM was also used by Chernova & Veloso (2007) to encode controllers for a Sony AIBO robot dog and a simulated driving task. One interesting feature of this work is that the system continuously evaluates the uncertainty

of the learned Gaussian mixture set and is able to stop and ask for further directions when the uncertainty is high, reducing the need for repetitive demonstrations of simple parts of the behavior.

de Rengervé et al. (2010) compared a GMM based encoding strategy with a Neural Network (NN) based controller, using a simple robot navigation task. The NN based controller appears to perform better with limited training data, and can give more direct feedback to the teacher. In contrast, the GMM does better in well known domains, when statistical features and variations can be properly estimated. The authors suggest that the two methods could be used complementarily, applying the NN during early learning sessions, with the GMM taking over when more training has been done.

2.3 Recognition

It is difficult to imagine a situation where one knows how to execute a certain behavior, but is unable to recognize someone else doing the same thing. However, for a robot, the ability to recognize behavior does not directly come with the ability to execute that behavior. Even though a controller generated through LFD may work well for executing the demonstrated behavior, it does not automatically provide a way to recognize that behavior.

A number of approaches to segmentation and recognition of behaviors can be found in the literature. Several measures have been proposed, including variance thresholding for certain sensor modalities (Peters II et al., 2003; Koenig & Matarić, 2006) and thresholding the mean velocity of joints (Fod et al., 2002; Nakaoka et al., 2003). Nicolescu (2003) recognizes behavior primitives by matching their pre- and postconditions with current sensory states. Support Vector Machines have been used for recognition of upper body postures (Ardizzone et al., 2000) and hand grasps (Zollner et al., 2002). Bentivegna (2004) uses a nearest-neighbor classifier on state data to identify skills in a marble maze task. Pook & Ballard (1993) present an approach where sliding windows of data are classified using Learning Vector Quantization (Kohonen, 2003) in combination with a nearest-neighbor classifier. *Hidden Markov Models (HMM)* are frequently used for recognition of gestures. One example is the work by Park et al. (2005) using a camera-based system to track the position of hands and head of a human, and an HMM for recognition of gestures based on hand and head positions. Fujie et al. (2004) use an HMM in a similar way, recognizing head gestures based on the optical flow in the visual scene.

In *Paper II*, we present and evaluate three additional techniques; β -comparison, AANN-comparison and S-comparison. β -comparison compares the outcome of a controller in response to the stimuli with observed actions in the demonstration. AANN-comparison is based on Autoassociative Neural Networks that model each skill such that the reconstruction error can be used for behavior recognition. Finally, S-comparison is based on S-Learning (Rohrer & Hulet, 2006b,a) and uses the sequence length as a measure of behavior similarity.

Even though several of these techniques work well for recognizing many types of skills, none provide a general solution to the problem. When seen as a classification

problem, behavior recognition appears ill posed. Which features of the demonstration that are relevant depends a lot on the particular behavior to be recognized. A demonstration may contain relevant features ranging from simple statistical properties to symbol-level conditions such as relations between objects (Calinon, 2009). Features that frequently appear in many demonstrations are often, but not always, more relevant. Overall, it appears that there is a significant need for task-specific biases also in the recognition process.

The problem of behavior recognition may be better approached by using information provided by the controller. One common way to do this is to implement a set of modules consisting of a controller π_M paired with a forward model ϕ_M , (e.g. Billard & Hayes, 1999; Demiris, 1999; Wolpert & Kawato, 1998; Haruno et al., 2001):

$$u_t = \pi_M(x_t) \tag{2.3}$$

$$\hat{y}_{t+1} = \phi_M(x_t, u_t) \tag{2.4}$$

where $x \in X_M$ is the state of module M , tuned for the particular behavior implemented by π_M . The forward model (predictor) computes the expected observation \hat{u}_{t+1} as a result of the action y_t taken in state x_t under the module's context. Simultaneously, the prediction \hat{u}_t , (made in previous time step $t - 1$) is compared to the actual observation u_t , producing a prediction error $\Delta_t = |u_t - \hat{u}_t|^2$. A large Δ_t indicates that the forward model is not tuned to the controller, or that observations do not correspond to the behavior implemented by π_M .

Successful behavior recognition using this approach requires that there be a module implementing the behavior to be recognized. However, the approach also supports learning of modules. Consider a set of modules initially implementing random forward and inverse models. When presented some data, one module will by chance receive the smallest prediction error and be selected as the responsible module. If the controller of this module is implemented as a mapping from observations to actions, it can be directly trained from a demonstration. Similarly, the predictor can be trained to minimize the prediction error for the presented data. As long as the mapping between observations and actions does not change, the active module will benefit from training and produce decreasing prediction errors, causing it to remain active. However, if large changes in world conditions appear, previous knowledge will no longer be applicable and the active module will perform worse than random modules. As a result, another module takes over responsibility and is tuned to the new conditions.

This mechanism is often put forward as an explanation of the role of the motor system in perception and imitation (e.g. Oztop et al., 2006; Rizzolatti & Craighero, 2004). As Demiris and Hayes put it:

The imitator is not imitating because it is understanding what the demonstrator is showing, but rather, it is understanding it because it is imitating. Imitation is used as a mechanism for bootstrapping further learning and understanding. (Demiris & Hayes, 2002)

Each module can be said to implement an internal simulation of sensor consequences in response to action, and action in response to current state. The module that produces the best prediction of observed events is selected as the best interpretation of observed events. That should be understood as one way to give the robot an inner world, a simulation of the physical world that does not rely on a pre-defined physics simulator but generated from interactions with the world. Such a simulation is inherently grounded in the robot's sensors and actuators and is consequently not subject to the symbol grounding problem (Harnad, 1990). A minimalistic implementation of this approach can be found in the work by Ziemke et al. (2005). This approach also has tight connections with the work by Barsalou and colleagues (e.g. Barsalou et al., 2003; Barsalou, 2009), describing the human cortex as a system simulating sensor percepts in relation to motor activity.

The use of a predictive measure to match the internal model with the actual world takes further support in that prediction error is proportional to the amount of free energy in the system (Friston & Stephan, 2007). Free energy is a thermodynamic measure describing the amount of work that a thermodynamic system can perform. In information theory, the term is used as a quantity that bounds the evidence of a model. For humans and other animals, the model is encoded by the brain and the data it models is the organism's interactions with the world. An organism that minimizes free energy will minimize the risk for unexpected exchanges with the environment, and consequently control entropy. This can be understood as an argument that any organism in the physical world must act to prevent surprises that can lead to potentially harmful states. Prediction error, with a specific measure of precision, is useful as a method for behavior recognition, not because forward models are very powerful classifiers, but because the purpose of acting, on a very fundamental level, is to minimize surprise (Friston, 2009).

CHAPTER 3

Hierarchical models for learning

This dissertation is a continuation of the work presented in the Licentiate thesis (Billing, 2009). The Licentiate thesis provides an analysis of several neurocomputational models of the brain (Riesenhuber & Poggio, 1999; Hawkins & Blakeslee, 2002; Haruno et al., 2003; Wolpert, 2003; Demiris & Simmons, 2006; George, 2008), all with some emphases on hierarchical structures and connections to the mirror system (Rizzolatti & Craighero, 2004). In an attempt to identify common features among these models, four criteria for general learning ability are proposed (Billing, 2009):

1. **Hierarchical structures**

Knowledge gained from learning should be represented in hierarchies.

2. **Functional specificity**

Knowledge gained from learning should be organized in functionally specialized modules.

3. **Forward and inverse**

Prediction error reflects how well the state definition satisfies the Markov assumption, and by consequence a forward model can be used to improve knowledge representation when paired with an inverse model.

4. **Bottom-up and top-down**

Both bottom-up and top-down signals must be propagated through the hierarchical structure. Bottom-up signals represent the state of modules, and top-down signals specify context.

Criteria 2 and 3 have already been discussed in the previous chapter, but criteria 1 and 4 may need some elaboration. In Section 3.1, some arguments for introducing hierarchies with the specific information flow given by Criterion 4 is presented. The argumentation leads up to a specific hierarchical architecture based on PSL, presented in Section 3.2.

3.1 Motivation for hierarchies

Hierarchical structures should be thought of as a useful and very general bias for representing knowledge about the physical world. Hierarchies are found almost everywhere in nature. Humans and animals consist of several body parts, which in turn consist of even smaller units down to cells and atoms. Scaling upwards, any organism lives in some kind of environment which can be viewed on many levels up to a scale where the earth is one component in an even greater ecosystem. The hierarchical structure of vegetables is even more apparent. Trees consists of branches which consist of even smaller branches which have leaves. Leaves have their own hierarchical structure which in fact resembles the tree itself in many ways. This apparent self-similarity of many natural structures has been extensively studied from a theoretical perspective as fractals, with works by Mandelbrot (1983) and Wolfram (2002) as prominent examples.

Hierarchies also exist in the temporal domain. Natural systems tend to have a nested organization with large-scale system variables and small-scale sub-system level variables. Large-scale system variables are often changing slower than the variables of its sub-systems Werner (1999). A common example of this temporal and spatial hierarchy can be drawn from weather. On a large scale, one can observe long term variations, such as seasons or even global warming. Simultaneously, there are local variations in the weather, such as storms and rain, which change much faster (George, 2008, p. 95).

Functional hierarchies appear to be a critical aspect of neural information processing, both spatially (Felleman & Van Essen, 1991; Hilgetag et al., 2000) and temporally (Boemio et al., 2005; Fuster, 2001). Further support comes from behavioral studies, for example the work by Byrne & Russon (1998) reporting hierarchical structure in voluntary behavior of great apes.

In a more technical respect, Criterion 1 can be motivated through an efficient division of labor between different parts of the system. A flat architecture implementing a set of functionally specific modules, as proposed in Section 2.3, relies only on forward models in order to control the switching between modules. Putting these modules into a hierarchy produces a system able to represent complex behaviors while keeping each module relatively simple. Modules at the bottom layer interacts directly with sensors and actuators of the robot at high temporal resolution. Modules higher up in the hierarchy have decreasing resolution, allowing these modules to efficiently express dependencies over longer periods of time. State variables that change slowly compared to a specific module's resolution are not included in the state description for that module, but are assumed to be provided by modules higher in up the hierarchy. This information is often referred to as the module's *context* (Wolpert & Ghahramani, 2000). Since modules higher up in the hierarchy are working at a lower temporal resolution, they can more easily capture slow variables, providing contextual information to lower modules. Conversely, variables that change quickly in comparison to the temporal resolution are handled lower in the hierarchy. This allows each module to be implemented as a semi-reactive controller, where the behavior depends on relatively recent states. Another advantage of this kind of architecture is that updates of a single

behavior or parts of a behavior will only require updates of a few modules and will not propagate changes to other parts of the system.

Wolpert & Ghahramani (2000) take the example of picking up a milk carton. The controller executing the behavior must take the amount of milk in the carton into consideration. It would be possible to consider the amount of milk as part of the module's state space, but that would pose a difficult sensing problem since the amount of milk is not directly visible. More importantly, since the amount of milk normally is constant during the operation of the module, it is not necessary to include as part of the state description. Instead, the amount of milk in the carton is treated as contextual information and handled by modules higher up in the hierarchy.

As the controller starts executing the lifting behavior, the forward model will produce predictions of expected sensor consequences of executed actions. If the expectation about the amount of milk is wrong, it will show up as large prediction errors when grasping the carton. This information is propagated upwards in the hierarchy, leading to a change in contextual information that better matches the actual amount of milk in the carton.

For this hierarchy of modules to be useful the system must be able to compute the system state at every level in the hierarchy. With the exception of modules at the bottom level (Layer 1), the input to each module is a context match for modules at the layer directly below. A context match should be understood as an estimate of how well present circumstances match the module's context, normally computed as a function of prediction error. The output of each module at Layer 2 and higher represents prior probabilities for each module in the layer below. This top-down information is often called a *responsibility signal* (e.g. Haruno et al., 2001) and is used both for control and training. During learning, only active modules are updated to fit present circumstances, while inactive modules remain unchanged. Ideally, this leads to a system where each module is in control only when the world satisfies its context. This interaction between layers in the hierarchy is one way to fulfill Criterion 4.

While the architecture presented here describes functional specificity as a set of distinct modules, it should be pointed out that this notion of module is quite different from how the term is used in many other robotic architectures, for example hybrid and deliberate systems (Murphy, 2000). These architectures often emphasize modularization in the sense that it should be possible to develop and test each module separately. This is not at all emphasized in the architecture presented here. In contrast, each module is to a high degree working in the environment constituted by the other modules and the overall behavior of the system is an emergent property of the interaction between modules, rather than a strict division of labor.

Criterion 3 was formulated to put emphasis on the functional specificity rather than on crisp modularization. An architecture based on crisp modules requires that each demonstrated sequence is interpreted as a specific sequence of selected modules, introducing a conflict between generalization and segmentation (Yamashita & Tani, 2008). This kind of architecture can potentially benefit from a partial overlap between modules. The important property of functional specificity is that errors identified under a certain context should not automatically propagate through the whole system, but only take effect within that particular context. A similar argument applied to re-

inforcement learning is put forward by Winberg & Balkenius (2007). Most work on reinforcement learning treats rewards and penalties as positive and negative changes of a single state value variable, respectively. Winberg and Balkenius demonstrate that learning time is decreased if rewards are generalized to similar states, while penalties are only applied to the current state. The reasoning behind this argument is that any over-generalizations of rewards will eventually be identified, since the agent will act to reach these states. In contrast, states with negative value will be avoided, causing any over-generalizations to remain.

One architecture able to share knowledge between different primitive behaviors is the recurrent neural network *RNNPB* (Tani et al., 2004). Both input and output layer of the network contain sensor and motor nodes, as well as nodes with recurrent connections. The input layer is given a set of extra nodes, representing *parametric bias* (PB). The network is trained to minimize prediction error, both by training the network using back-propagation and by changing the PB input vector. The PB vector is however updated slowly, such that it organizes into what could be seen as a context layer for the rest of the network. In addition to giving the network the ability to represent different behaviors that share knowledge, the PB vector can be used for behavior recognition. In a continuation of this work, Yamashita & Tani (2008) demonstrate the emergence of a functional hierarchy in an architecture based on a *continuous time recurrent neural network* (CTRNN). Context nodes (nodes that are not connected to inputs or outputs) of the network are given different time constants. Through training, slow nodes drive switching between behaviors, while fast nodes express dynamics within each behavior.

While the present work is directed to methods for robot learning, similar arguments can be found in several fields of research. The four criteria presented above embody the assumption that a wide range of cognitive phenomena can be explained through one, or a few, basic mechanisms. As mentioned in the previous section, Friston (2009) uses the notion of free-energy to describe such a fundamental mechanism, applying to any living organism. In a less mathematical formulation, Hawkins & Blakeslee (2002) introduce the notion of a *common cortical algorithm* as a basic computational description of the brain. *Activity Theory* (Kaptelinin & Nardi, 2006) describes high level cognitive function, including consciousness, as inherently grounded in action and learning as a progressive internalization of the physical world leading to hierarchical representations. A longer discussion on the brain as a prospective organ is found in the work by Sjölie (2011).

3.2 Hierarchical predictive learning

Based on the four criteria outlined in the previous section, this section describes a general purpose architecture for LFD. The architecture presented here is not fully implemented and evaluated, but builds on the results from Papers IV to VII. A central part of this architecture is the PSL algorithm. The algorithm treats control as a prediction problem, such that the next action is selected based on the sequence of recent sensory-motor events. In addition, PSL also produce predictions of expected sensor

states. While these are not directly useful for control, predictions of sensor states appear to serve well as a method for behavior recognition, as proposed in Section 2.3.

PSL has many interesting properties as a learning algorithm for robots. It is model free, meaning that it introduces very few assumptions into learning and needs little task specific configuration. PSL can be seen as a variable-order Markov model. Starting out from a reactive (first-order) model, PSL estimates transition probabilities between sensory-motor states. For states that do not display the Markov property, the order is increased and PSL models the transition probability based on several passed events. In this way, PSL will progressively gain memory for the parts of the behavior that cannot be modeled in a reactive way. Knowledge is stored as hypotheses h , describing a statistical relation between a sequence of passed events $(e_{t-|h|+1}, e_{t-|h|+2}, \dots, e_t)$, and a future event e_{t+1} :

$$h : (e_{t-|h|+1}, e_{t-|h|+2}, \dots, e_t) \xrightarrow{C(h)} e_{t+1} \quad (3.1)$$

where $C(h)$ is the confidence of h and $|h|$ denotes the length of h , i.e., the number of left-hand-side conditions. In Paper VII, we present a version of PSL where $C(h)$ is context dependent. A hypothesis that is fairly weak in the general case can in this way still receive high confidence in a particular context. Given a number of contexts C , defined as fuzzy sets over all h , the belief in each context $\lambda_t(C)$ at time t is given by Bayes' rule:

$$\lambda_t(C) = \frac{\lambda_{t-1}(C) \exp\left(\frac{|e_t - \hat{e}_t^C|^2}{2\sigma^2}\right)}{\sum_{i=1}^N \left[\lambda_{t-1}(C_i) \exp\left(\frac{|e_t - \hat{e}_t^{C_i}|^2}{2\sigma^2}\right) \right]} \quad (3.2)$$

where N is the number of contexts and \hat{e}_t^C is the prediction produced by PSL within context C . The variance σ^2 is used as a scaling constant controlling the size of confidence changes in relation to prediction error size.

$\lambda_t(C)$ can be used to directly control which context is to be responsible at a certain time (Paper VII). Transitions between contexts may however also be controlled by another instance of PSL, running at a context level (Figure 3.1). In each time step, PSL queries the knowledge base selecting the best hypothesis based on how well each hypothesis matches observed events, and respective confidence $C(h)$. At the context level, a similar computation is made, selecting a context hypothesis h' based on the sequence of transitions between contexts. The instance of PSL running at the context level handles data with lower temporal resolution, allowing context hypotheses with relatively few left hand side conditions to stretch over fairly long periods of time.

While PSL at the observation level works directly with the sequence of sensory-motor events, the context layer works with information that can not be predicted at the observation level (Figure 3.2). Sensory-motor events that can be correctly predicted by the observation layer does not contain any new information, and is therefore filtered away. Conversely, prediction error represents new information that could not be explained by the observation layer, and is therefore regarded as more valuable. This

argument relates to the discussion in Section 2.3. A successful prediction at the context level will reduce prediction errors at the observation level, producing a system that acts to reduce predictions at both levels of abstraction.

In Paper VII, we use manually defined contexts trained from demonstrations of different behaviors. However, the teacher's interpretation of what is one behavior, and what is another, does not necessarily correspond to the robot's need to categorize experiences. It may consequently be an advantage not to rely on manually defined contexts. Automatic identification of suitable contexts could possibly be formulated as an optimization problem where the total entropy of the model, over all contexts, is to be minimized.

If a context layer can be organized dynamically, it opens the possibility to add a second context layer that interacts with the first context layer in a similar way that the first context layer interacts with the observation layer (Figure 3.2). Potentially, the hierarchy could be extended even further, to create a fully hierarchical system minimizing prediction errors at many levels of abstraction.

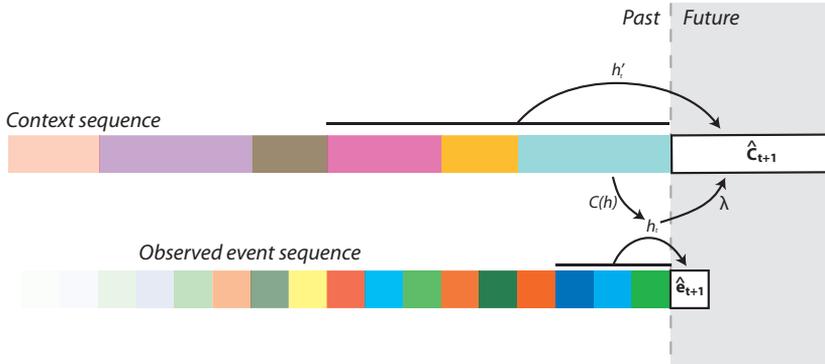


Figure 3.1: Interaction between PSL at observation level and PSL at context level. h is the selected hypothesis at observation level and h' is a hypothesis selected by PSL running at context level. \hat{e}_{t+1} is the predicted event at observation level and \hat{C}_{t+1} represents predicted context responsibility.

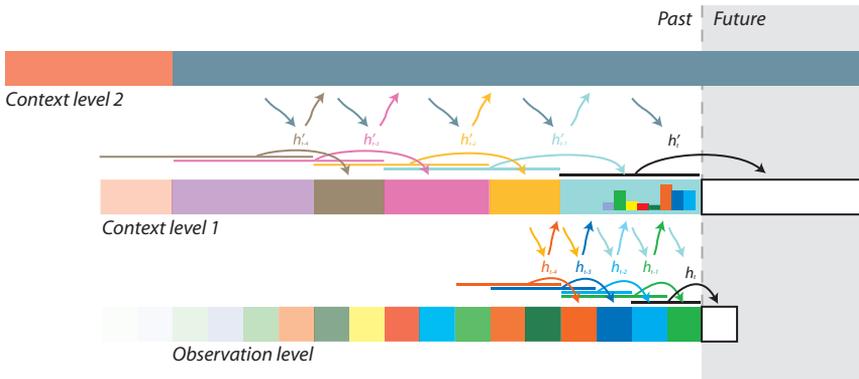


Figure 3.2: Three layer architecture with one instance of PSL is running at each level.

CHAPTER 4

Summary of articles

Starting out from a cognitive perspective, Paper I reviews the problem of robot learning and representation of behavioral knowledge in a broad sense. Several robotic paradigms are compared and put in relation with the theory of *extended mind* (Clark & Chalmers, 1998), *distributed cognition* (Hutchins, 1995) and *situated action* (Suchman, 1987). The paper was an early attempt to identify how these cognitive theories apply to robotics. The work recognizes the importance of separating the agent's perspective from the perspective of the observer and argues that many of the fundamental problems of artificial intelligence can be solved if the robot's knowledge is represented as a coordination between sensors and actuators, rather than as an explicit world model. At the same time, notions of emergent behavior and self-organization, that are often brought up in response to traditional representations, are criticized since they rarely provide a solution to the kind of problems that explicit representations do.

Paper II presents an evaluation of several techniques for behavior recognition. The use of behavior primitives with behavior recognition is one way to use previous knowledge in order to generalize the demonstration such that the behavior can be executed under varying conditions. While behavior recognition is possible to implement for particular behaviors or classes of behaviors, it appears as a very difficult problem in the general case. One reason for this may lie in a vague definition of behavior and lack of general bias for selecting one interpretation over another. Large parts of this dissertation argues for the use of a predictive measure in order to approach the problem of behavior recognition in a general way.

Paper III presents a formalization of LFD, specifically directed to LFD when using behavior primitives. Central concepts in LFD are defined, including behavior, demonstration, sensors and actuators. A possibility to use behavior primitives as a way to interpret and display a demonstration for a human user is also explored. A visualization of the robot's interpretation of the behavior could serve both as a way for the human user to evaluate the effect of learning, and as a way to give feedback to the robot.

Papers IV to VII present and evaluate a specific technique for LFD called *Predictive Sequence Learning (PSL)*. In Paper IV, PSL is used as a controller for a miniature Khepera robot (K-Team, 2007). The result from this work indicates that PSL is applicable as a controller for simpler behaviors. PSL is however unable to correctly reproduce more complex behaviors. The reason for this is that knowledge of one part of the behavior may interfere with knowledge of other parts, causing PSL to mix up situ-

ations and select inappropriate actions. This problem, called knowledge interference, could potentially be solved by integrating PSL with a high-level controller that selects which part of the PSL knowledge base that should be active in a certain situation. One such high-level controller, based on semantic networks, is proposed by Fonooni et al. (2012). Parts of the PSL knowledge base is in this architecture integrated as nodes in the semantic network.

In Paper V, an evaluation of two ways to apply PSL as a method for behavior recognition is presented, including a comparison with previous work from Paper II. The evaluation shows that PSL is able to recognize previously learned behaviors in a demonstration. The recognition is based on the same model as PSL uses for control. If this results holds in the general case, PSL could potentially constitute one component in a dynamic hierarchical system similar to the one presented in Chapter 3.

In Paper VI, a new version of PSL based on Fuzzy Logic is presented and compared to the original algorithm. Fuzzy PSL handles dimensionality better than the original, discrete, version of the algorithm. This allows us to scale up the problem to a Robosoft Kompai robot (Robosoft, 2011) acting in a simulated apartment environment. Paper VII extends this work by integrating on-line behavior recognition with the PSL-based controller. We are able to reproduce the results from both Paper IV and V using Fuzzy PSL and the new evaluation environment. The on-line behavior recognition is used to compute a responsibility signal, controlling which part of the PSL knowledge base that is active at a certain point in time. In this way, the problem of knowledge interference appears more manageable.

CHAPTER 5

Contributions

The primary contribution of this dissertation lies in the design and evaluation of *Predictive Sequence Learning (PSL)* (Paper IV to VII). The algorithm has a straight forward design that introduces the right kind of biases for general learning ability in robots. Two important attributes are the semi-reactive behavior and a progressive extension of the knowledge base. The semi-reactive behavior could probably be captured just as well by many other algorithms, for example a recurrent neural network. The progressive extending knowledge base is less common and reduces the risk for catastrophic forgetting present in many other approaches, including neural networks. On the negative side, PSL in its current form can not cope with high-dimensional data in a good way. Investigating how this could be solved could be one focus for a continuation of this work. Improved handling of high-dimensional data could possibly be achieved in a similar way that the combinatorial explosion in the temporal dimension is handled through the use of contexts (Paper VII). While we have shown that PSL does work as controller and as a method for behavior recognition in some fairly realistic robotic applications, much more work is needed to evaluate the algorithm. Comparisons with a recurrent neural network such as RNNPB (Tani et al., 2004) or Dynamic Field Theory (Amari, 1977) could be interesting directions for future work.

In Paper III, we present a formalization of LFD. This formalism itself is not revolutionary in any way, to large extent it adapts descriptions that have been standard in the field for a long time. However, the paper describes important problems of LFD, especially with a focus on introduction of bias in learning, in a novel way.

Another contribution of this dissertation is the hierarchical architecture for general purpose LFD presented in Section 3.2 that builds on four criteria for general learning ability originally presented in the Licentiate thesis (Billing, 2009). In many respects, the presented architecture is not new, similar approaches have been around in the literature for more than ten years. However, the present work adds new properties by emphasizing semi-reactive modules and by proposing a concrete system allowing partial knowledge overlap between functionally specific modules. One interesting direction for future work is to evaluate how human notions of behavior correspond to the hierarchical system's need to categorize information into contexts on different levels. Above all, I hope that the present work approaches the question of general machine learning in a realistic way. Through the work with this dissertation I have been convinced that it is possible to create computer systems able to learn in a similar way that humans and other animals do. I believe that this line of research can support develop-

ment of useful robots and other machines, but also contribute to a better understanding of human cognition.

Bibliography

- Alissandrakis, A., Nehaniv, C. L., & Dautenhahn, K. (2002). Imitation With ALICE: Learning to Imitate Corresponding Actions Across Dissimilar Embodiments. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 32, 482–496.
- Alissandrakis, A., Nehaniv, C. L., Dautenhahn, K., & Saunders, J. (2005). An Approach for Programming Robots by Demonstration: Generalization Across Different Initial Configurations of Manipulated Objects. In *Proceedings of 2005 International Symposium on Computational Intelligence in Robotics and Automation* (pp. 61–66). Espoo, Finland.
- Amari, S. (1977). Dynamics of Pattern Formation in Lateral-Inhibition Type Neural Fields. *Biological Cybernetics*, 27(2), 77–87.
- Ardizzone, E., Chella, A., & Pirrone, R. (2000). Pose classification using support vector machines. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium* (pp. 317–322 vol.6).
- Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5), 469–483.
- Arkin, R. C. (1998). *Behaviour-Based Robotics*. MIT Press.
- Barsalou, L. W. (2009). Simulation, situated conceptualization, and prediction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521), 1281–1289.
- Barsalou, L. W., Simmons, K. W., Barbey, A. K., & Wilson, C. D. (2003). Grounding conceptual knowledge in modality-specific systems. *Trends in Cognitive Sciences*, 7(2), 84–91.
- Bentivegna, D. C. (2004). *Learning from Observation using Primitives*. PhD thesis, Georgia Institute of Technology.
- Berk, L. E. & Winsler, A. (1995). *Scaffolding Children's Learning: Vygotsky and Early Childhood Education*. National Association for the Education of You.
- Berthouze, L. & Metta, G. (2005). Epigenetic robotics: modelling cognitive development in robotic systems. *Cognitive Systems Research*, 6(3), 189–192.

- Billard, A., Epars, Y., Cheng, G., & Schaal, S. (2003). Discovering imitation strategies through categorization of multi-dimensional data. In *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, volume 3 (pp. 2398–2403 vol.3). Las Vegas, Nevada.
- Billard, A. & Hayes, G. (1999). DRAMA, a Connectionist Architecture for Control and Learning in Autonomous Robots. *Adaptive Behavior*, 7(1), 35–63.
- Billing, E. A. (2009). *Cognition Reversed - Robot Learning from Demonstration*. Licentiate thesis, Umeå University, Department of Computing Science, Umeå, Sweden.
- Bischoff, R. & Guhl, T. (2009). Robotic Vision to 2020 and Beyond - The Strategic Research Agenda for Robotics in Europe.
- Boemio, A., Fromm, S., Braun, A., & Poeppel, D. (2005). Hierarchical and asymmetric temporal sensitivity in human auditory cortices. *Nature neuroscience*, 8(3), 389–95.
- Byrne, R. W. & Russon, A. E. (1998). Learning by Imitation: A Hierarchical Approach. *The Journal of Behavioral and Brain Sciences*, 16(3).
- Calinon, S. (2009). *Robot Programming by Demonstration - A Probabilistic Approach*. EFPL Press.
- Calinon, S. & Billard, A. (2005). Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM. In *Proceedings of the International Conference on Machine Learning (ICML)* (pp. 105–112). Bonn, Germany: ACM.
- Calinon, S., Guenter, F., & 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.
- Chernova, S. & Veloso, M. (2007). Confidence-based policy learning from demonstration using Gaussian mixture models. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems - AAMAS '07* (pp. 1315–1322). New York, New York, USA: ACM Press.
- Clark, A. & Chalmers, D. J. (1998). The Extended Mind. *Analysis*, 58, 10–23.
- de Rengervé, A., D'halluin, F., Andry, P., Gaussier, P., & Billard, A. (2010). A study of two complementary encoding strategies based on learning by demonstration for autonomous navigation task. In *Proceedings of the Tenth International Conference on Epigenetic Robotics* Lund, Sweden.
- Demiris, J. & Hayes, G. (1997). Do robots ape? In *Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents* (pp. 28–31).

- Demiris, J. & Hayes, G. R. (2002). Imitation as a dual-route process featuring predictive and learning components: A biologically plausible computational model. In K. Dautenhahn & C. L. Nehaniv (Eds.), *Imitation in animals and artifacts* (pp. 327–361). Cambridge, MA, USA: MIT Press.
- Demiris, Y. (1999). *Movement Imitation Mechanisms in Robots and Humans*. PhD thesis, University of Edinburgh.
- Demiris, Y. & Simmons, G. (2006). Perceiving the unusual: Temporal properties of hierarchical motor representations for action perception. *Neural Networks*, 19(3), 272–284.
- Felleman, D. J. & Van Essen, D. C. (1991). Distributed Hierarchical Processing in the Primate Cerebral Cortex. *Cereb Cortex*, 1, 1–17.
- Fod, A., Matarić, M., & Jenkins, O. C. (2002). Automated derivation of primitives for movement classification. *Autonomous Robots*, (pp. 39–54).
- Fonooni, B., Hellström, T., & Janlert, L. E. (2012). Learning High-Level Behaviors from Demonstration through Semantic Networks. In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART) (to appear)* Vilamoura, Algarve, Portugal.
- Friston, K. J. (2009). The free-energy principle: a rough guide to the brain? *Trends in Cognitive Sciences*, 13(7), 293–301.
- Friston, K. J. & Stephan, K. E. (2007). Free-energy and the brain. *Synthese*, 159(3), 458, 417.
- Fujie, S., Ejiri, Y., Nakajima, K., Matsusaka, Y., & Kobayashi, T. (2004). A conversation robot using head gesture recognition as para-linguistic information. In *13th IEEE International Workshop on Robot and Human Interactive Communication (ROMAN 2004)*. (pp. 159–164).
- Fuster, J. M. (2001). The Prefrontal Cortex - An Update. *Neuron*, 30(2), 319–333.
- George, D. (2008). *How the Brain might work: A Hierarchical and Temporal Model for Learning and Recognition*. PhD thesis, Stanford University.
- Harnad, S. (1990). The Symbol Grounding Problem. *Physica*, D(42), 335–346.
- Haruno, M., Wolpert, D. M., & Kawato, M. (2001). Mosaic model for sensorimotor learning and control. *Neural computation*, 13(10), 2201–2220.
- Haruno, M., Wolpert, D. M., & Kawato, M. (2003). Hierarchical MOSAIC for movement generation. In *International Congress Series 1250* (pp. 575– 590).: Elsevier Science B.V.
- Hawkins, J. & Blakeslee, S. (2002). *On Intelligence*. Times Books.

- Hilgetag, C. C., O'Neill, M. A., & Young, M. P. (2000). Hierarchical organization of macaque and cat cortical sensory systems explored with a novel network processor. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 355(1393), 71–89.
- Hovland, G., Sikka, P., & McCarragher, B. (1996). Skill acquisition from human demonstration using a hidden Markov model. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3 (pp. 2706–2711).
- Hutchins, E. (1995). *Cognition in the Wild*. Cambridge, Massachusetts: MIT Press.
- K-Team (2007). Khepera robot. www.k-team.com.
- Kaptelinin, V. & Nardi, B. A. (2006). *Acting with Technology: Activity Theory and Interaction Design*. The MIT Press.
- Koenig, N. & Matarić, M. J. (2006). Behavior-Based Segmentation of Demonstrated Tasks. In *Proceedings of the International Conference on Development and Learning* Bloomington, USA.
- Kohonen, T. K. (2003). Learning vector quantization. In M. A. Arbib (Ed.), *The Handbook of Brain Theory and Neural Networks* (pp. 631–638). MIT Press.
- Mandelbrot, B. B. (1983). *The Fractal Geometry of Nature*. W. H. Freeman.
- Matarić, M. J. (2002). Sensory-motor primitives as a basis for imitation: linking perception to action and biology to robotics. In *Imitation in animals and artifacts* (pp. 391–422). MIT Press.
- Murphy, R. R. (2000). *Introduction to AI Robotics*. MIT Press.
- Myers, B. C. S. & Rabiner, L. R. (1981). A Comparative Study of Several Dynamic Time-Warping. *The Bell System Technical Journal*, 60(7), 1389–1409.
- Nakaoka, S., Nakazawa, A., Yokoi, K., & Ikeuchi, K. (2003). Recognition and generation of leg primitive motions for dance imitation by a humanoid robot. In *Proceedings of 2nd International Symposium on Adaptive Motion of Animals and Machines* Kyoto, Japan.
- Nehaniv, C. L. & Dautenhahn, K. (1999). Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications. *Interdisciplinary Approaches to Robot Learning*, 24, 136–161.
- Nehaniv, C. L. & Dautenhahn, K. (2001). Like Me? - Measures of Correspondence and Imitation. *Cybernetics and Systems*, 32, 11–51.
- Nicolescu, M. (2003). *A Framework for Learning from Demonstration, Generalization and Practice in Human-Robot Domains*. PhD thesis, University of Southern California.

- Otero, N., Saunders, J., Dautenhahn, K., & Nehaniv, C. L. (2008). Teaching robot companions: the role of scaffolding and event structuring. *Connection Science*, 20, 111–134.
- Oztop, E., Kawato, M., & Arbib, M. (2006). Mirror neurons and imitation: a computationally guided review. *Neural networks : the official journal of the International Neural Network Society*, 19(3), 254–71.
- Pal Robotics (2011). REEM Humanoid Service Robot. www.pal-robotics.com.
- Park, H., Kim, E., Jang, S., Park, S., Park, M., Kim, H., Marques, J., Pérez de la Blanca, N., & Pina, P. (2005). HMM-Based Gesture Recognition for Robot Control. In J. S. Marques, N. Pérez de la Blanca, & P. Pina (Eds.), *LNCS Pattern Recognition and Image Analysis*, volume 3522 of *Lecture Notes in Computer Science* (pp. 695–716). Berlin, Heidelberg: Springer Verlag.
- Peters II, R. A., Campbell, C. L., Bluethman, W. J., & Huber, E. (2003). Robonaut Task Learning through Teleoperation. In *Proceedings of the 2003 IEEE Intl. Conference on Intelligent Robots and Automation* (pp. 23–27). Taipei, Taiwan.
- Pook, P. K. & Ballard, D. H. (1993). Recognizing Teleoperated Manipulations. In *Proceedings on 1993 IEEE International Conference on Robotics and Automation, 1993*. (pp. 578–585).
- Riesenhuber, M. & Poggio, T. (1999). Hierarchical Models of Object Recognition in Cortex. *Nature Neuroscience*, 2(11), 1019–25.
- Rizzolatti, G. & Craighero, L. (2004). The Mirror-Neuron System. *Annual Review of Neuroscience*, 27, 169–192.
- Robosoft (2011). Kompai Robot, www.robosoft.com.
- Rohrer, B. & Hulet, S. (2006a). A learning and control approach based on the human neuromotor system. In *Proceedings of the First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanics*. (pp. 57–61).
- Rohrer, B. & Hulet, S. (2006b). *BECCA - A Brain Emulating Cognition and Control Architecture*. Technical report, Cybernetic Systems Integration Department, Sandria National Laboratories, Albuquerque, NM, USA.
- Sjölie, D. (2011). *Reality-based brain-computer interaction*. Licentiate thesis, Department of Computing Science, Umeå University, Umeå, Sweden.
- Suchman, L. A. (1987). *Plans and Situated Actions*. Cambridge University Press.
- Tani, J., Ito, M., & Sugita, Y. (2004). Self-Organization of Distributedly Represented Multiple Behavior Schemata in a Mirror System : Reviews of Robot Experiments Using RNNPB. *Neural Networks*, 17, 1273–1289.
- Taylor III, A. (2011). Is Google Motors the new GM? *CNN Money*.

- Thrun, S. & Pratt, L. Y., Eds. (1998). *Learning to Learn*. Kluwer Academic Publishers.
- Vygotsky, L. S. (1978). *Mind in Society: Development of Higher Psychological Processes*. Harvard University Press, 14th edition.
- Werner, B. T. (1999). Complexity in Natural Landform Patterns. *Science*, 284(5411), 102–104.
- Winberg, S. & Balkenius, C. (2007). Generalization and Specialization in Reinforcement Learning. In L. Berthouze, C. G. Prince, M. Littman, H. Kozima, & C. Balkenius (Eds.), *Proceedings of the Seventh International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*. Lund, Sweden.
- Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media, 1 edition.
- Wolpert, D. H. & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82.
- Wolpert, D. M. (2003). A unifying computational framework for motor control and social interaction. *Phil. Trans. R. Soc. Lond.*, B(358), 593–602.
- Wolpert, D. M. & Ghahramani, Z. (2000). Computational principles of movement neuroscience. *Nature Neuroscience*, 3, 1212–1217.
- Wolpert, D. M. & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7-8), 1317–1329.
- Yamashita, Y. & Tani, J. (2008). Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. *PLoS computational biology*, 4(11), e1000220.
- Ziemke, T., Jirnhed, D. A., & Hesslow, G. (2005). Internal simulation of perception: a minimal neuro-robotic model. *Neurocomputing*, 68, 85–104.
- Zollner, R., Rogalla, O., Dillmann, R., & Zollner, M. (2002). Understanding users intention: programming fine manipulation tasks by demonstration. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and System*, volume 2 (pp. 1114–1119).